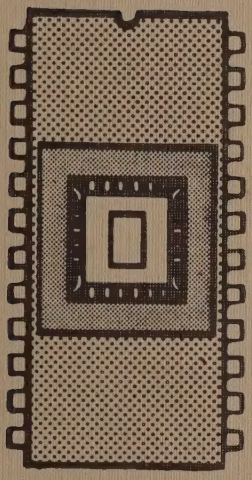
LOGICAL DEVICES, INC.



PROMPRO 7

EPROM PROGRAMMING UNIT

INSTRUCTION MANUAL

LOGICAL

LOGICAL DEVICES, INC.

IMPORTANT NOTICE

Every effort has been made to make the supplied documentation as accurate as possible, however, Logical Devices Inc. will not assume responsibility for any damages incurred by such material. Logical Devices Inc. reserves the right to changes in such material at any time.

PLEASE CHECK FOR CHANGE INFORMATION AT THE REAR OF THIS MANUAL.

PROMPRO 7

EPROM PROGRAMMING UNIT

INSTRUCTION MANUAL

LOGICAL DEVICES, INC.
1321E N.W. 65 PLACE
FT. LAUDERDALE, FLORIDA 33309

Serial Number _____

TABLE OF CONTENTS

SECTION	SUBJECT	PAGE
1.0	UNPACKING	1
2.0	INTRODUCTION	2
2.1	STAND-ALONE MODE	2
2.2	TERMINAL MODE	4
2.3	COMPUTER MODE	5
3.0	COMMAND DICTIONARY	7
4.0	STATUS LED DISPLAYS	11
5.0	DEVICE CONFIGURATION	12
5.1	DEVICE TABLES	13
6.0	SERIAL RS-232 INTERFACE	15
6.1	SERIAL RS-232 BAUD RATE	16
7.0	SOFTWARE INTERFACE	17
7.1	USING CPM	20
8.0	TROUBLESHOOTING GUIDE	23
9.0	COMMUNICATION PROBLEMS	24
10.0	CALIBRATION GUIDE	27
A	APPENDIX A. HEX FORMATS	
B	APPENDIX B. SOFTWARE DRIVERS	
C	APPENDIX C. SERIAL CONNECTIONS	
D	APPENDIX D. EPROMs WHAT THEY ARE	
E	APPENDIX E. HOW TO INSERT EPROMs	
F	APPENDIX F. XP7 FIRMWARE OPTION	

TABLE OF CONTENTS

SECTION	SUBJECT	PAGE
1.0	INTRODUCTION	1
2.0	[illegible]	2
3.0	[illegible]	3
4.0	[illegible]	4
5.0	[illegible]	5
6.0	[illegible]	7
7.0	[illegible]	11
8.0	[illegible]	12
9.0	[illegible]	13
10.0	[illegible]	15
	[illegible]	16
	[illegible]	17
	[illegible]	20
	[illegible]	23
	[illegible]	24
	[illegible]	27
A	[illegible]	
B	[illegible]	
C	[illegible]	
D	[illegible]	
E	[illegible]	
F	[illegible]	

1.0 UNPACKING

1) CAREFULLY UNPACK UNIT FROM THE SHIPPING CONTAINER AND INSPECT FOR POSSIBLE SHIPPING DAMAGE.

2) CHECK TO SEE IF THE FOLLOWING ITEMS HAVE BEEN INCLUDED:

A) PROMPRO-7 EPROM PROGRAMMER
B) SET OF 6 CONFIGURATORS
C) FEMALE DB-25 CONNECTOR
D) FUSE 1/2 AMP
E) MANUAL
F) WARRANTY REGISTRATION

3) INSPECT FUSE HOLDER, INSERT 1/2A FUSE.

4) TOGGLE THE ON/OFF SWITCH TO OFF POSITION AND PLUG UNIT INTO THE THE AC OUTLET.
THE ON/OFF SWITCH IS LOCATED IN THE REAR OF THE UNIT.

5) TURN THE POWER SWITCH TO THE ON POSITION. THE FRONT PANEL STATUS LED'S SHOULD LIGHT IN SEQUENCE (RED ERROR, YELLOW BUSY, THEN GREEN READY).
THE GREEN LED WILL STAY ON TO INDICATE THAT THE UNIT IS READY TO ACCEPT A COMMAND.

6) NOTICE SMALL SQUARE BLACK FOAM ON THE FRONT PANEL IS TO PLACE YOUR MOST COMMONLY USED CONFIGURATORS.

2.0 INTRODUCTION

THE PROMPRO-7 PROGRAMMER IS A STAND-ALONE UNIT COMPLETE WITH CABINET AND POWER SUPPLY. IT IS DESIGNED TO BE USED WITH A MINIMUM AMOUNT OF EFFORT. USER CAN BECOME FAMILIAR WITH OPERATION OF THE UNIT IN JUST A FEW SHORT MINUTES. THERE ARE THREE BASIC MODES OF OPERATIONS: STAND-ALONE, TERMINAL CONTROLLED, AND COMPUTER CONTROLLED.

2.1 STAND ALONE MODE

IN THIS MODE OF OPERATION USER CAN COPY AND VERIFY EPROMS BY THE USE OF THE FRONT PANEL FUNCTION SWITCHES. THERE IS NO NEED FOR A COMPUTER OR TERMINAL HOOK-UP. JUST FOLLOW THESE FEW SIMPLE INSTRUCTIONS:

1) TURN THE PROMPRO-7 POWER SWITCH ON IN THE BACK OF THE UNIT.

2) INSERT THE PROPER PIN CONFIGURATOR IN THE CNF SOCKET WITH THE WHITE DOT IN THE DIRECTION OF THE ZIF SOCKETS.
REFER TO SECTION 5.1 TO CHOOSE THE PROPER CNF.

3) INSERT MASTER EPROM IN THE CORRESPONDING SOCKET. PIN 1 OF THE EPROM IS ON THE SIDE OF THE ZIF HANDLE. REFER TO APPENDIX E ON HOW TO INSERT EPROMS .

4) DEPRESS THE FRONT PANEL RESET SWITCH. THIS WILL RESET THE SYSTEM AND AUTOMATICALLY LOAD THE ENTIRE CONTENTS OF THE EPROM IN THE RAM BUFFER (YOU WILL SEE THE LEDS GO ON IN SEQUENCE AS IN THE POWER-UP PROCEDURE).

5) REMOVE THE MASTER EPROM AND REPLACE WITH AN ERASED EPROM (ALWAYS WORKING FROM THE SAME ZIF SOCKET FOR THE SAME TYPE OF EPROM).
DO NOT ATTEMPT TO PLUG IN ANY OF THE 24 PIN DEVICES IN THE SKT2 SOCKET.
REFER TO SECTION 5.0 FOR SOCKET SELECTION.

6) BE SURE THAT THE EPROM IS PROPERLY ERASED. SEE NOTE AT THE END OF THIS SECTION.

7) CHECK TO ENSURE THAT THE EPROM AND THE CONFIGURATORS ARE INSERTED IN THE SOCKET WITH THEIR PROPER ORIENTATION. THIS IS A VERY IMPORTANT STEP TO PREVENT ANY DAMAGE TO THE DEVICE.

C A U T I O N

DO NOT PLUG IN CONFIGURATOR BACKWARD. YOU MAY DAMAGE YOUR DEVICE.

8) DEPRESS THE PROG SWITCH ON THE FRONT PANEL. THE BUSY (YELLOW L.E.D.) INDICATOR WILL REMAIN ON FOR THE PROGRAMMING PERIOD. ONCE THE PROGRAMMING IS COMPLETED, THE READY (GREEN L.E.D.) WILL BE ACTIVATED. IF THE EPROM DOES NOT PROGRAM PROPERLY, THE ERROR (RED L.E.D.) INDICATOR WILL GO ON.
(AN AUDIBLE ALARM IS ALSO PROVIDED TO ALERT THE USER FOR AN OVERCURRENT OR AN ERROR CONDITION.)

9) TO INSURE THAT EVERY LOCATION HAS BEEN PROPERLY PROGRAMMED, THE VERIFY FEATURE CAN BE UTILIZED BY DEPRESSING THE VERIFY SWITCH ON THE FRONT PANEL. THIS COMMAND WILL INITIATE A COMPARISON OF EVERY DATA BYTE IN THE INTERNAL RAM AGAINST THE EPROM. IF THERE IS ANY DISCREPANCY, THE RED ERROR L.E.D. WILL GO ON.

N O T E S

TO BE POSITIVELY SURE THAT AN EPROM IS PROPERLY ERASED:

A. WITH A KNOWN BLANK EPROM INSERTED IN THE UNIT, PRESS THE RESET SWITCH;
B. INSERT THE EPROM IN QUESTION INTO THE UNIT;
C. PRESS THE VERIFY SWITCH. IF THE RED L.E.D. GOES ON, THE EPROM IS NOT COMPLETELY ERASED.

WARNING

CERTAIN DEFECTIVE EPROMS MAY CAUSE A DIRECT SHORT FROM VPP TO GROUND INTERNALLY. THIS WILL CAUSE THE PROMPRO-7 TO GO INTO CURRENT LIMIT. SHOULD THIS PROBLEM ARISE YOU MAY NEED TO TURN THE UNIT OFF FOR SEVERAL SECONDS BEFORE OPERATING AGAIN. EXCESSIVE HEAT GENERATED BY THE EPROM DURING THE PROGRAMMING CYCLE IS A GOOD INDICATION OF SUCH PROBLEM. REFER TO CALIBRATION PROCEDURE TO CHECK VPP.

2.2 TERMINAL MODE

THE PROMPRO-7 HAS THE ABILITY TO OPERATE DIRECTLY WITH A TERMINAL. AN INTERNAL COMMUNICATIONS PROGRAM ALLOWS USER TO UTILIZE A TERMINAL AS A PERIPHERAL TO THE PROMPRO-7.

1) CONNECT THE UNIT TO A SERIAL TERMINAL OR A COMPUTER THROUGH THE RS-232 CONNECTOR IN THE BACK OF THE UNIT. REFER TO THE TABLE BELOW FOR THE CONNECTOR PIN ASSIGNMENT:

PROMPRO-7 RS-232 PIN	SIGNAL
PIN 1	CHASSIS GROUND
PIN 2	RECEIVE
PIN 3	TRANSMIT
PIN 4	RTS (NOT USED)
PIN 5	CTS
PIN 6, 20 (LOOPED)	DTR, DSR
PIN 7	SIGNAL GROUND

SEE SECTION 6.0 FOR BAUD RATE SETTING.
NOTE: CTS IS A HANDSHAKE LINE THAT IS NOT REQUIRED TO BE USED IN THE TERMINAL MODE. THIS LINE SHOULD BE LEFT DISCONNECTED.
REFER TO SECTIONS 6.0 FOR MORE DETAILS.

2) THE PROGRAMMER TRANSMITS A COMMAND MENU FOLLOWED BY A PROMPT "*" UPON POWER-UP OR RESET. A COMMAND IS ENTERED AFTER THE PROMPT.

```
E(X)AM
INTEL
MOTOROLA
PROGRAM
TRANSFER
UPLOAD
VERIFY
*
```

3) WHEN THE UNIT IS RESET A MENU WILL BE PRINTED OUT ON THE TERMINAL, FOLLOWED BY AN ASTERISK. THIS IS THE PROMPT USED TO INDICATE THAT THE UNIT IS READY TO ACCEPT A COMMAND.

4) AFTER THE COMPLETION OF ANY TASK THE PROMPRO-7 WILL TRANSMIT THE CHARACTERS "▪F*EOJ*" OR "▪UEOJ*". THE CHARACTER "▪" PRECEDING ANY CHARACTER INDICATES THE CONTROL CODE FOR THAT CHARACTER. THUS, A "▪F" (HEX 06) WILL INDICATE AN ACK (SHORT FORM FOR ACKNOWLEDGE) MEANING THAT ALL WENT WELL IN THE PREVIOUS INSTRUCTION. ON THE OTHER HAND, IF A "▪U" (HEX 15) IS TRANSMITTED, THIS WILL INDICATE A NAK (SHORT FORM FOR NEGATIVE ACKNOWLEDGE) MEANING THAT AN ERROR WAS ENCOUNTERED THE PREVIOUS INSTRUCTION. THE CODES $06 OR $15 ARE TYPICALLY NON-GENERATING CHARACTERS THAT WOULD ONLY BE USED WHEN CONNECTED TO A HOST COMPUTER.

NOTE: "▪U*EOJ*" TRANSMISSION SEQUENCE MEANS THAT THE CONTROL U FOLLOWED BY ASCII CHARACTERS *EOJ* IS SENT.

2.3 COMPUTER MODE

THIS MODE OF OPERATION IS SIMILAR TO THE TERMINAL MODE, WITH THE DIFFERENCE THAT USER CAN NOW TRANSFER HEX FILES TO AND FROM THE PROMPRO-7. THE MAXIMUM RAM BUFFER SIZE ON THE PROMPRO-7 IS 32K BITS, OR 4K BYTES. FILES LONGER THAN THIS SIZE WILL WRAP-AROUND.
THE COMPUTER MODE THEN, IN ESSENCE, CONSISTS OF TWO DISTINCT MODES OF OPERATIONS:
A. TERMINAL MODE- IN THIS MODE OF OPERATION THE COMPUTER IS MADE TO LOOK LIKE A TERMINAL TO THE OUTSIDE PERIPHERAL (PROGRAMMER IN THIS CASE). USER THE CAN OPERATE THE PROMPRO-7 AS DESCRIBED IN THE SECTION 2.2.
TO MAKE YOUR COMPUTER LOOK LIKE A TERMINAL YOU MUST INSTRUCT YOUR COMPUTER VIA A SOFTWARE PROGRAM. THIS TYPE OF SOFTWARE PROGRAM IS GENERALLY REFERRED TO AS A "TERMINAL" OR "MODEM" PROGRAM AND IS GENERALLY AVAILABLE COMMERCIALLY UNDER VARIOUS NAMES; "CROSS-TALK" , "MODEM-7" ARE SOME EXAMPLES.

NOTE: THE SERIAL CABLING TO A COMPUTER IS GENERALLY DIFFERENT THAN TO A CRT TERMINAL. REFER TO SECTION 6.0.

B. FILE TRANSFER MODE- THIS MODE OF OPERATION ALLOWS USER TO TRANSFER A HEX FILE TO THE PROGRAMMER OR READ A HEX FILE FROM THE PROGRAMMER.
THE FORMAT OF THE THESE HEX FILES ARE SHOWN IN THE FOLLOWING PAGES.

THE FOLLOWING IS THE ACTUAL SEQUENCE OF ASCII CHARACTERS SENT DOWN TO THE PROGRAMMER THROUGH SERIAL RS-232 PORT:

```
I----------T-:1000000027040D.............
|    |     |  |____ INTEL HEX FILE
|    |     |_______ DOWNLOAD COMMAND
|    |_____________ 11 CHARACTER TIME FOR PROMPRO-7
|                   TO RETURN ACK CHARACTERS:
|                     <CR><LF>*F*EOJ*<CR><LF>*
|__________________ SPECIFY INTEL FORMAT
```

NOTE: THE CHARACTER TIME IS NEEDED BECAUSE PROMPRO-7 COMMUNICATES IN THE HALF DUPLEX MODE: THAT IS, IT CAN ONLY TRANSMIT OR RECEIVE AT ONE TIME. THEREFORE, WHILE IT IS TRANSMITTING THE ACKNOWLEDGE CHARACTERS, IT WILL NOT BE ABLE TO SEE ANY CHARACTERS TRANSMITTED TO IT BY THE COMPUTER. YOU CAN USE THE ASCII CHARACTER "T" TO FILL IN FOR CHARACTER TIME. FOR FURTHER DETAILS REFER TO SECTION 7.0 AND 7.1.
YOU WILL FIND AN ASCII CHART IN APPENDIX B

FOR UPLOAD:

```
U1----------:100100000027040D.............
|                 |____ INTEL HEX FILE RETURNED
|                       BY PROMPRO-7 AFTER
|                     RECEIVING U1 AND 5 SEC. DELAY
|_____________________ UPLOAD COMMAND WITH ADDRESS
                       STRARTING AT 1000 HEX
```

3.0 PROMPRO-7 COMMAND DICTIONARY

3.1 RESET/LOAD (RESET PUSHBUTTON)

THE RESET FUNTION IS DONE ON BOTH POWER-UP OR BY PRESSING THE RESET BUTTON. THIS FUNCTION ALSO TRANSFERS THE CONTENTS OF THE EPROM TO THE RAM.

3.2 EXAMINE MEMORY CONTENTS X HHHH

THE EXAM COMMAND ALLOWS THE USER TO GAIN DIRECT ACCESS TO A SPECIFIC MEMORY LOCATION, AND IF DESIRED, CHANGE THE CONTENTS OF THAT THE LOCATION.

A. TO USE THIS COMMAND TYPE "X".
B. A SPACE WILL THEN BE GENERATED.
C. THE UNIT IS NOW WAITING FOR A 4 DIGIT HEX ADDRESS. THE FIRST DIGIT WILL ALWAYS BE INTERPRETED AS A ZERO REGARDLESS IF A ZERO WAS ENTERED OR NOT. THIS IS BECAUSE THE RAM HAS AN ADDRESS RANGE FROM 0000 TO OFFF.
D. AFTER THE ADDRESS HAS BEEN ENTERED, THE UNIT WILL GENERATE THE ADDRESS, AN "=" SIGN, AND THE CONTENTS OF THIS LOCATION.

E. IF NO CHANGE IS TO BE MADE TO THIS LOCATION A "SPACE" WILL INCREMENT THE ADDRESS AND DISPLAY THE CONTENTS OF THE NEXT LOCATION. TO CHANGE DATA, SIMPLY ENTER NEW DATA (TWO DIGIT HEX).

F. TO TERMINATE THE EXAM COMMAND, TYPE A "CR".

EXAMPLE: EXAMINE LOCATION 0200 AND CHANGE
LOCATION 0201 FROM 67 TO 55
VERIFY THAT LOCATION 201 CONTAINS 55.

```
*X 0200
0200=45 <SPACEBAR>
0201=67 <55><CR>
*
*X 0201
0201=55 <CR>
*
```

3.3 INTEL HEXADECIMAL FORM (I)

AN "I" ENTERED VIA THE RS-232C PORT WILL DIRECT ALL FILE TRANSFER COMMUNICATIONS TO BE CARRIED OUT IN THE INTEL HEXADECIMAL FORMAT.
SEE APPENDIX A FOR DETAILS AND EXAMPLES.

3.4 MOTOROLA HEXADECIMAL FORMAT (M)

AN "M" ENTERED VIA THE RS-232C PORT WILL DIRECT ALL COMMUNICATIONS TO BE CARRIED OUT IN THE MOTOROLA HEXADECIMAL FORMAT. SEE APPENDIX A.

3.5 FORMATTED HEXADECIMAL UP-LOAD (U H)

WHEN DIRECTED, THE UNIT WILL TRANSMIT THE CONTENTS OF THE RAM. THE AMOUNT THAT WILL BE TRANSMITTED WILL BE DETERMINED BY THE TYPE OF EPROM AND THE CONFIGURATOR.

AN EXAMPLE OF THIS WOULD BE:

2716=2K
2532=4K
2732=4K
8748=1K
8749=2K

A. TO COMMAND THE UNIT TO ENTER THE UP-LOAD MODE, TYPE A "U".

B. THE UNIT IS NOW EXPECTING A SECOND CHARACTER. THIS SECOND CHARACTER WILL BE A HEXADECIMAL HALF-BYTE, OFFSETTING THE STARTING ADDRESS OF THE TRANSMISSION. THE PROMPRO-7 HAS A 4K (4096) BYTE BUFFER WHICH WOULD BE AN ACTUAL ADDRESS RANGE OF 0000-0FFF, BUT YOUR RECEIVING BUFFER MAY NOT BE IN THIS ADDRESS RANGE. IT MUST BE POINTED OUT THAT THIS IS STRICKLY A HALF-BYTE OFFSET WHICH ONLY ALLOWS THE USER TO AFFECT THE MOST SIGNIFICANT HALF-BYTE OF THE ADDRESS.

EXAMPLE:
THE FOLLOWING IS AN EXAMPLE OF A 4K BYTE UP-LOAD FROM A 2732.

CHARACTERS RECEIVED BY PROGRAMMER	STARTING ADDRESS	FINAL ADDRESS
U0	0000	0FFF
U3	3000	3FFF
U9	9000	9FFF
UC	C000	CFFF

C. THE UNIT WILL NOW TRANSMIT THE CONTENTS OF THE RAM IN THE HEXADECIMAL FORMAT PREVIOUSLY SPECIFIED. IF NO FORMAT IS SPECIFIED THE DEFAULT FORMAT WILL BE MOTOROLA. A 7 SECOND DELAY BEFORE TRANSMISSION IS INCORPORATED IN ORDER TO ALLOW SETUP TIME.

3.6 FORMATTED HEXADECIMAL DOWN-LOAD (T)

A. TO COMMAND THE UNIT TO ENTER THE DOWN-LOAD MODE, TYPE A "T".

B. THE UNIT IS NOW EXPECTING THE HEADER CHARACTER OF THE SPECIFIED FORMAT, AN "S" FOR MOTOROLA, OR AN ":" FOR INTEL. AT THIS POINT ALL OTHER CHARACTERS WILL BE IGNORED UNTIL THE HEADER IS RECEIVED.

C. THE MOST SIGNIFICANT NIBBLE OF THE ADDRESS WILL BE STRIPPED IN ORDER TO ACCOMODATE THE 4K ADDRESS RANGE OF THE UNIT (0000-0FFF).

D. IF NO CHARACTER IS RECEIVED WITHIN APPROXIMATELY 7 SECONDS, THE UNIT WILL EXIT THE DOWN-LOAD MODE AND RETURN TO THE READY STATE.

E. THE CHECKSUM IS COMPARED AT THE END OF EACH DATA RECORD. IF AN ERROR IS DETECTED THE ERROR L.E.D. WILL LIGHT BUT THE DATA WILL STILL BE STORED IN RAM AND THE UNIT WILL STILL REMAIN IN THIS MODE UNTIL THE END OF THE RECORD IS FOUND OR UNTIL TRANSMISSION STOPS.

3.7 PROGRAM PROM (P) (PROG PUSHBUTTON)

WHEN THE UNIT IS DIRECTED TO PROGRAM THE EPROM, A SIMPLE TRANSFER TAKES PLACE; THE CONTENTS OF RAM ARE BURNED IN TO THE COINCIDING LOCATIONS OF THE EPROM. THE EPROM MUST HAVE BEEN PREVIOUSLY ERASED BY THE "QUV-T8 EPROM ERASER" OR EQUIVALENT.

A. THE FIRST METHOD OF ENTERING THE PROGRAMMING MODE IS BY SENDING A "P" TO THE UNIT VIA THE RS-232C PORT.
B. THE SECOND METHOD OF ENTERING THE PROGRAMMING MODE IS BY PRESSING THE PROGRAM BUTTON.
C. AFTER THE EPROM HAS BEEN COMPLETELY PROGRAMMED, ALL LOCATIONS ARE THEN COMPARED TO THE RAM. IF ANY ERRORS ARE FOUND, THE ERROR L.E.D. WILL LIGHT.
D. IF AN ERROR OCCURS, CHECK THE CONFIGURATOR FOR PROPER TYPE AND COMPLETE INSERTION.

W A R N I N G

DO NOT ATTEMPT TO PROGRAM TWO DEVICES AT THE SAME TIME. BOTH DEVICES MAY BE DESTROYED.

3.8 VERIFY (V)(VER PUSHBUTTON)

THIS FEATURE, WHETHER DIRECTED FROM THE TERMINAL OR THE FRONT PANEL SWITCH, INITIATES A LOCATION BY LOCATION COMPARISON OF THE RAM BUFFER AGAINST THE EPROM.
IF A DISCREPANCY OCCURS, THE RED LED AND THE AUDIBLE ALARM WILL GO ON. IF ALL IS WELL THE GREEN LED GOES ON. THIS IS A VERY USEFUL FEATURE FOR COMPARING A LARGE NUMBER OF EPROMS AGAINST A KNOWN MASTER.

4.0 STATUS LED DISPLAYS

THERE ARE THREE LED INDICATORS ON THE FRONT PANEL OF THE UNIT USED EITHER FOR STAND-ALONE MODE OR RS-232 MODE. THE FOLLOWING DESCRIBES THE FUNCTIONS OF THE DISPLAYS:

4.1 ERROR L.E.D. (RED)

THIS L.E.D. WILL LIGHT WHEN THE FOLLOWING FAULTS OCCUR:

A. AN EPROM THAT DID NOT PROGRAM CORRECTLY. TRY ERASING THE EPROM AGAIN FOR A LONGER PERIOD OF TIME. ALSO, CHECK THE CONFIGURATOR.

B. AN ERROR WAS FOUND WHEN PERFORMING A DOWN-LOAD. CHECK BAUD-RATE SELECTION. CHECK FOR NON-VALID HEX ENTRY.

C. OVER-CURRENT INDICATION ON PROGRAM. IN THIS CASE THE RED LED WILL GO ON SOLID IMMEDIATELY AFTER THE OVER-CURRENT CONDITION OCCURS. THE AUDIBLE INDICATOR WILL ALSO GO ON.

4.2 BUSY L.E.D. (YELLOW)

THIS L.E.D. WILL LIGHT WHEN EVER ANY ACTIVE FUNCTION IS CURRENTLY ENGAGED.

W A R N I N G

DO NOT ATTEMPT TO INSTALL OR REMOVE AN EPROM ANY TIME THAT THIS L.E.D. IS ON. DEVICE DESTRUCTION MAY OCCUR.

4.3 READY L.E.D. (GREEN)

THIS L.E.D. WILL LIGHT WHEN THE UNIT IS IN IDLE STATE. ALL POWER, ADDRESS, AND DATA LINES HAVE BEEN TAKEN LOW (I.E. LESS THAN 1 VOLT) MAKING IT SAFE TO REMOVE OR INSTALL THE EPROM AT THIS TIME.

5.0 DEVICE CONFIGURATION

DEVICE TYPE IS SELECTED. IN ORDER TO CONFIGURE THE PROGRAMMER FOR A PARTICULAR DEVICE TYPE USER MUST PLUG INTO THE FRONT PANEL CNF SOCKET ONE OF THE SMALL DIP CONFIGURATOR PACKAGES SUPPLIED AT THE TIME OF PURCHASE.

THESE PACKAGES ARE CODED WITH NUMBERS 1 THROUGH 8. EACH NUMBER REPRESENTING ONE OR MORE TYPE OF DEVICES.

NOTE: CERTAIN DEVICES WILL REQUIRE A PERSONALITY ADAPTOR AS WELL AS A CONFIGURATOR. THE FOLLOWING TABLE WILL LIST ALL OF THE DEVICES THAT ARE SUPPORTED DIRECTLY AND THE ONES THAT WILL REQUIRE EXTERNAL ADAPTOR MODULES.

TO USE THE DEVICE TABLE SIMPLY LOCATE THE TYPE OF DEVICE OR THE EQUIVALENT ON THE LEFT COLUMN. THEN DETERMINE THE CONFIGURATOR NUMBER YOU MUST INSERT IN THE CNF SOCKET UNDER THE CNF COLUMN. IF THE DEVICE IS ONE THAT IS SUPPORTED AS STANDARD FEATURE THEN YOU CAN INSERT YOUR CHIP DIRECTLY IN ONE OF THE TWO SKT SOCKETS INDICATED UNDER THE SOCKET COLUMN. IF THE DEVICE REQUIRES AN OPTIONAL ADAPTOR, YOU MUST FIRST PLUG-IN THE ADAPTOR IN ONE OF THE ZIF SOCKETS INDICATED UNDER THE SOCKET COLUMN AND THEN INSERT YOUR PROM IN THE ADAPTOR ZIF SOCKET.

USER CAN CHANGE CONFIGURATORS WHILE POWER IS ON WITHOUT DAMAGE TO THE UNIT. IT IS POSSIBLE TO EDIT RAM WITHOUT THE CONFIGURATORS. IT IS NOT NECESSARY TO HAVE AN EPROM OR A CONFIGURATOR IN THE SOCKET WHILE IN THE RS-232 OPERATION MODE. USER CAN TRANSFER DATA FROM ONE TYPE OF EPROM TO ANOTHER TYPE BY CHANGING CONFIGURATORS AFTER THE LOAD OPERATION.
IF YOU LOSE YOUR CONFIGURATOR YOU CAN ALWAYS OBTAIN MORE FROM THE FACTORY. DO NOT INSERT ANY OTHER TYPE OF HARDWARE IN THE CNF SOCKET AS IT MAY DAMAGE THE SOCKET.

5.1 DEVICE DICTIONARY

THE PROMPRO-7 CAN SUPPORT THE FOLLOWING DEVICES:

DEVICE SIZE	DEVICE TYPE	CNF #	STANDARD SUPPORT	ADAPTER OPTION	SOCKET (ZIF)
ADVANCED MICRO DEVICES (AMD):					
2K X 8 :	AM2716DC, AM2716NDC (ALL)	#1	YES	-	SKT1
4K X 8 :	AM2732	#3	YES	-	SKT1
8K X 8 :	AM2764	#7	-	PM28	SKT1
FAIRCHILD:					
4K X 8 :	F2732	#3	YES	-	SKT1
FUJITSU:					
4K X 8 :	MBM2732	#3	YES	-	SKT1
8K X 8 :	MBM2764	#7	-	PM28	SKT1
8K X 8 :	MBM27C64	#7	-	PM28	SKT1
HITACHI:					
2K X 8 :	HN462716G	#1	YES	-	SKT1
4K X 8 :	HN462732G	#3	YES	-	SKT1
4K X 8 :	HN462732AG	#4	YES	-	SKT1
8K X 8 :	HN482764G	#7	-	PM28	SKT1
INTEL CORP.:					
1K X 8 :	8748	#1	YES	-	SKT2
1K X 8 :	8748H	#6	YES	-	SKT2
1K X 8 :	8741	#1	-	PM41	SKT2
1K X 8 :	2758A/B	#1	YES	-	SKT1
2K X 8 :	2816 EEPROM	#1	-	PM26	SKT1
2K X 8 :	2716	#1	YES	-	SKT1
2K X 8 :	8742	#5	-	PM41	SKT2
2K X 8 :	8749H	#5	YES	-	SKT2
2K X 8 :	8755A	#8	-	PM55	SKT2
4K X 8 :	2732	#3	YES	-	SKT1
4K X 8 :	2732A	#4	YES	-	SKT1
4K X 8 :	8751	#7	-	PM51	SKT1
8K X 8 :	2764	#7	-	PM28	SKT1
16K X 8 :	27128	#7	-	PM28	SKT1
MITSUBISHI:					
2K X 8 :	M5L2716K	#1	YES	-	SKT1
4K X 8 :	M5L2732K	#3	YES	-	SKT1
MOSTEK:					
2K X 8 :	MK2716-N (ALL)	#1	YES	-	SKT1

DEVICE SIZE	DEVICE TYPE	CNF #	STANDARD SUPPORT	ADAPTER OPTION	SOCKET (ZIF)
MOTOROLA:					
2K X 8 :	MCM2716, MCM27L16	#1	YES	-	SKT1
4K X 8 :	MCM2532	#2	YES	-	SKT1
NATIONAL SEMICONDUCTOR:					
2K X 8 :	MM2716E, MM2716M, NMC27C16	#1	YES	-	SKT1
4K X 8 :	NMC2732	#3	YES	-	SKT1
8K X 8 :	MM27C64	#7	-	PM28	SKT1
TEXAS INSTRUMENTS:					
2K X 8 :	TMS 2516	#1	YES	-	SKT1
4K X 8 :	TMS 2532	#2	YES	-	SKT1

CAUTION---8748/49 USERS

INSERT 8748/49 IC IN SKT SOCKET FOR LOAD, VERIFY, AND PROGRAM OPERATIONS ONLY. REMOVE IC FROM SOCKET WHENEVER USING OTHER COMMANDS SUCH AS MEMORY EXAM AND EXCHANGE, OR FILE TRANSFER. THIS IS A SAFETY PRECAUTION TO PREVENT ANY POSSIBLE DAMAGE TO THESE IC's.

6.0 SERIAL INTERFACE

PROMPRO-7 HAS A BUILT-IN SERIAL I/O. THE SERIAL CONNECTOR IS ACCESSIBLE THROUGH DB-25 MALE CONNECTOR LOCATED IN THE REAR OF THE UNIT, SEE APPENDIX C.

THE PIN-OUT FOR THE SERIAL I/O IS SUCH THAT IT CAN BE DIRECTLY CONNECTED TO A CRT TERMINAL (WITH PIN 5 LEFT DISCONNECTED). FOR MOST COMPUTER INTERFACES THE PINS 2 AND 3 MUST BE REVERSED. IT IS ALWAYS A GOOD IDEA TO REFER TO THE PIN-OUT OF THE SERIAL CARD FOR YOUR TERMINAL OR COMPUTER JUST TO MAKE SURE THAT YOUR WIRING IS CORRECT.

RS-232C PIN ASSIGNMENT

PROMPRO-7 RS-232 PIN#

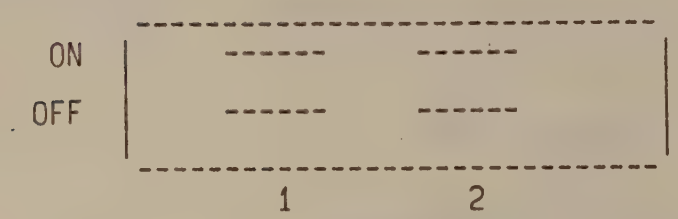
1	CHASSIS GROUND
2	RX DATA
3	TX DATA
4	NOT USED (RTS)
5	CLEAR TO SEND(INPUT)
6	DATA SET READY(NOT USED)
7	SIGNAL GROUND
20	DATA TERMINAL READY(NOT USED)

NOTE: PROMPRO-7 PROVIDES AN INTERNAL PULL-UP RESISTOR FOR THE CTS LINE. USER HAS THE OPTION TO EXERCISE THIS HANDSHAKE LINE. CERTAIN DEVELOPMENT SYSTEMS MAY REQUIRE LONG DELAYS IN RESPONDING TO THE SERIAL PORT, IN WHICH CASE THE CTS LINE TO PROMPRO-7 MUST BE USED. IF CTS LINE IS NOT ACTIVATED BY THE SERIAL PORT, MAKE SURE TO DISCONNECT PIN 5 FROM THE CABLE TO AVOID ACCIDENTLY HANGING UP THIS HANDSHAKE LINE.

SETTING THE BAUD RATE

PRIOR TO ESTABLISHING COMMUNICATIONS BETWEEN A CRT TERMINAL OR A COMPUTER AND THE PROMPRO-7, THE BAUD RATE SWITCH ON THE FRONT PANEL SHOULD BE SET TO THE SAME BAUD RATE AS YOUR TERMINAL OR COMPUTER:

1	2	BAUD
ON	ON	110
OFF	ON	300
ON	OFF	1200
OFF	OFF	2400



2-POSITION DIP SWITCH (FRONT PANEL TOP VIEW)

SERIAL DATA FORMAT:

HALF-DUPLEX
8-DATA BITS
ONE START BIT
TWO STOP BITS
PARITY IGNORED

SERIAL SIGNAL LEVELS: +12V,-12V (CTS ON +12V, NOT CTS ON -12V)

7.0 SOFTWARE INTERFACE

IN ORDER TO ESTABLISH COMMUNICATION BETWEEN PROMPRO-7 AND A COMPUTER, A COMPUTER RS-232C SERIAL PORT (110, 300, 1200, OR 2400 BAUD) MUST BE PROVIDED.
IN A TYPICAL DEVELOPMENT CYCLE USER ASSEMBLES PROGRAM INTO AN OBJECT FILE AND THEN TRANSFERS THE OBJECT FILE FROM THE MEMORY OR DISK TO THE PROMPRO-7.
SOME SOFTWARE DEVELOPMENT TOOLS SUCH AS ASSEMBLERS AND COMPILERS ALREADY GENERATE THE OUTPUT OF THE ASSEMBLY OR IN THE FORM OF INTEL OR MOTOROLA HEX FORMATS. IN SUCH A CASE THE JOB OF SENDING A FILE TO THE PROM PROGRAMMER IS ONLY A MATTER OF INSTRUCTING THE COMPUTER TO TRANSFER THE FILE FROM THE DISK(SOURCE) TO THE SERIAL PORT (DESTINATION).
MOST OPERATING SYSTEMS PROVIDE A UTILITY COMMAND TO FACILITATE FILE TRANSFER OPERATIONS. FOR EXAMPLE, UNDER THE CPM OPERATING SYSTEM, THE "PIP" COMMAND IS USED TO DO SUCH A TASK. UNDER PC DOS YOU MAY USE THE COPY COMMAND.
IF YOUR DEVELOPMENT SYSTEM DOES NOT GENERATE THE STANDARD FORMATS, THEN YOU MUST PROVIDE A PROGRAM TO CONVERT A BINARY OR STRAIGHT ASCII FILE TO THE STANDARD HEX FORMATS.
THE FOLLOWING SOFTWARE DRIVERS ARE AVAILABLE FROM LOGICAL DEVICES INC. ON DISC :

IBM PC SOFTWARE DRIVER
APPLE -II SOFTWARE DRIVER
INTEL MDS SOFTWARE DRIVER
CPM 2.2 8" SS SD
FLEX 09 5" SS SD
TEKTRONICS 8002

SOFTWARE DRIVERS CONSIST OF TWO DISTINCT PARTS;
1- A SIMPLE COMMUNICATIONS (TERMINAL) MODE.
2- HEX CONVERTER MODE.
THE ABOVE MENTIONED SOFTWARE DRIVERS HAVE ONE OR BOTH PARTS INCORPORATED INTO THEM.
FOR THE MOST PART, YOU MAY NOT EVEN NEED A SOFTWARE DRIVER SHOULD YOUR SYSTEM HAVE A TERMINAL MODE OR TERMINAL PROGRAM. FOR EXAMPLE, THE APPLEII SOFTWARE DRIVERS DO NOT HAVE TERMINAL MODE BECAUSE THE APPLEII COMPUTER CAN ALREADY ACT AS A TERMINAL WITH A FEW SIMPLE CONTROL CHARACTERS.

IN THE APPENDIX A YOU WILL FIND THE LISTING OF SEVERAL TYPES OF MICROPROCESSORS FOR GENERATING THE INTEL OR MOTOROLA HEX FORMATS. IN SOME INSTANCES THE LISTING PROVIDED(6809 ROUTINES) IS PART OF A LARGER PROGRAM . IT IS PROVIDED TO GIVE YOU ONLY AN ILLUSTRATION OF HOW IT IS DONE. THERE ARE ALSO EXAMPLES OF SOFTWARE DRIVERS FOR THE COMMODORE-64 AND THE TRS-80 COLOR COMPUTER. YOU CAN USE THESE ROUTINES AS A GUIDE TO DEVELOP YOUR OWN FOR ANY OTHER COMPUTER SYSTEM (SEE APPENDIX B)
BEFORE YOU ATTEMPT TO ANALYZE THE PROGRAM CODE IT IS A GOOD IDEA TO KNOW WHAT THE GENERAL STRUCTURE OF THE SOFTWARE PROGRAM SHOULD LOOK LIKE.

THE FOLLOWING DEFINES THE SOFTWARE STRUCTURE FOR THE DRIVERS(A SLIGHTLY MODIFIED VERSION OF A STANDARD TERMINAL PROGRAM).

1. INITIALIZE SERIAL PORT.
2. SCAN SERIAL PORT FOR INPUT.
3. IF INPUT, DISPLAY ON TERMINAL.
4. SCAN TERMINAL FOR USER ENTRY.
5. IF NO ENTRY, GO TO STEP 2.
6. OUTPUT USER ENTRY TO SERIAL PORT.
7. IF USER ENTRY IS "T" GO TO DOWNLOAD ROUTINE.
8. IF USER ENTRY IS "U" GO TO UPLOAD ROUTINE, OR GO TO STEP 2.

DOWNLOAD ROUTINE

1. PROMPT USER WITH THE FILE NAME OR MEMORY ADDRESSES WHERE THE DATA IS LOCATED.

1. SEND EITHER THE "I" OR "M" COMMAND TO SET THE PROGRAMMER IN THE PROPER HEX MODE.

3. SEND ASCII "T" (54) TO THE SERIAL PORT TO INDICATE PROGRAMMER THAT THE NEXT ENTRY IS GOING TO BE A VALID HEX FILE.

4. SEND HEX FILE TO THE SERIAL PORT.

UPLOAD ROUTINE
1. PROMPT USER FOR DISK FILE NAME OR MEMORY ADDRESS WHERE THE UPLOADED DATA IS TO BE STORED.
2. SEND TO THE SERIAL PORT ASCII "U" FOLLOWED BY A SECOND CHARACTER(0-F) INDICATING THE 4K OFFSET FOR THE LOAD ADDRESS IN SYSTEM RAM.

3. GO TO HEX LOAD ROUTINE.
4. RETURN.

NOTE: YOUR DRIVER ROUTINE ADDRESS SPACE MUST NOT COINCIDE WITH THE SAME ADDRESS SPACE AS YOUR UPLOAD FILE.

IT IS VERY IMPORTANT TO INSURE THAT YOUR SERIAL INTERFACE IS WORKING PROPERLY. IF YOU ARE USING IT FOR THE FIRST TIME THERE MAY BE A CHANCE THAT YOU MAY HAVE A PROBLEM WITH IT. READ THE INSTRUCTION MANUAL FOR YOUR SERIAL PORT VERY CAREFULLY. MONITOR THE TRANSMIT AND RECEIVE LINE WITH AN OSCILOSCOPE TO INSURE THAT DATA IS GOING OUT OR COMING IN. IN DEBUGGING ANY SYSTEM ALWAYS ISLOLATE THE UNKNOWNS AND DEAL WITH ONE UNKNOWN AT A TIME.

IT IS ALSO NECESSARY TO UNDERSTAND HOW THE SERIAL COMMUNICATION IN YOUR PARTICULAR COMPUTER IS ACCOMPLISHED.

EACH SYSTEM UTILIZES A SLIGHTLY DIFFERENT HARDWARE FOR THE SERIAL/MODEM PORT. FOR MANY SYSTEMS A SIMPLE "TERMINAL/MODEM" PROGRAM IS ALL THAT IS REQUIRED TO OPERATE THE PROMPRO-7 IN THE CONVERSATIONAL MODE. THESE PROGRAMS ARE READILY AVAILABLE FOR MOST SYSTEMS. A TERMINAL PROGRAM IS BASICALLY A PROGRAM THAT MAKES YOUR COMPUTER LOOK LIKE A CRT TERMINAL. ALL CONSOLE ENTRIES ARE DIRECTED TO THE SERIAL PORT AND ALL DATA RECEIVED FROM YOUR SERIAL PORT IS DISPLAYED ON THE SCREEN. PRIOR TO SENDING AND RECEIVING DATA TO AND FROM THE SERIAL PORT YOU MUST SET THE FOLLOWING PARAMETERS:

START BITS	1
STOP BITS	2
NO. DATA BITS	8
PARITY	IGNORED
BAUD	110, 300, 1200, 2400
MODE	HALF DUPLEX

THE ABOVE FORMAT IS SOFTWARE SELECTABLE IN CERTAIN SYSTEMS. FOR EXAMPLE, IN A RADIO-SHACK TRS-80 MODEL II WITH CPM 2.2 DOS, YOU CAN UTILIZE THE "SETUP"COMMAND TO SET YOUR PARAMETERS. IF YOU DO NOT HAVE A TERMINAL PROGRAM AND WISH TO WRITE YOUR OWN, THERE ARE THREE BASIC WAYS TO WRITE SUCH PROGRAMS.

1. TERMINAL PROGRAM WRITTEN IN BASIC LANGUAGE.
2. TERMINAL PROGRAM USING THE CALLABLE ROUTINES IN BIOS OR SYSTEM MONITOR.
3. ASSEMBLY LANGUAGE PROGRAM WITH YOUR OWN SERIAL PORT DRIVERS.

ALL COMMUNICATION IS DONE IN ASCII FORMAT. FOR EXAMPLE. IF YOU WERE TO DIRECT THE PROGRAMMER TO PROGRAM AN EPROM, YOU WOULD STORE THE ASCII EQUIVALENT OF "P" (50) IN THE DATA REGISTER OF YOUR SERIAL CONTROLLER.

THE I/O ADDRESSES IN MANY SYSTEMS ARE MEMORY MAPPED. THAT IS: TO TRANSFER A BYTE TO AND FROM THE I/O PORT WOULD BE SIMILAR TO TRANSFERRING A BYTE TO AND FROM A MEMORY LOCATION.
IF YOUR SYSTEM IS NOT I/O MEMORY MAPPED, YOU MUST USE THE INPUT/OUTPUT INSTRUCTIONS OF YOUR MICROPROCESSOR WITH THE CORRESPONDING I/O# ASSIGNMENT. MOST SERIAL CONTROLLER CHIPS HAVE TWO INTERNAL REGISTERS, ONE DATA AND ONE CONTROL/STATUS REGISTER. IF YOU ARE WRITING DIRECTLY TO THE SERIAL CONTROLLER YOU MUST ALWAYS TEST THE CONDITION OF THE READY STATUS BIT TO INSURE THAT YOU DO NOT OVER-RUN PREVIOUS DATA.

7.1 USING CPM

SINCE CPM IS THE MOST COMMON OPERATING SYSTEM FOR THE MICROCOMPUTERS SOME EXAMPLES OF ITS USE WITH THE PROMPRO-7 IS PROVIDED HERE:
WE ASSUME THAT YOU HAVE WRITTEN OR PURCHASED A TERMINAL PROGRAM FOR YOUR CPM SYSTEM BY THE NAME OF "MODEM.COM"

ONCE YOU POWER UP YOUR SYSTEM YOU SHOULD GET THE CPM PROMPT:

```
A>
                    [NOW TYPE "MODEM" AND RETURN]
A>MODEM <CR>
```

ONCE THE PROGRAM IS ACTIVATED IT WILL PROMPT YOU FOR SEVERAL OPTIONS. SELECT THE "TERMINAL" OPTION.
IF YOUR PROMPRO-7 IS CONNECTED PROPERLY TO THE SERIAL PORT, AS SOON AS YOU DEPRESS THE RESET SWITCH ON THE FRONT PANEL OF PROMPRO-7, THE FOLLOWING MENU SHOULD APPEAR ON YOUR SCREEN:

```
E(X)AM
INTEL
MOTOROLA
PROGRAM
TRANSFER
UPLOAD
VERIFY
*
```

TO SET THE PROGRAMMER IN THE INTEL HEXADECIMAL MODE, TYPE "I"

```
*I
```

PROMPRO-7 WILL RESPOND WITH:

```
*EOJ*
*
```

TO EXAMINE AND CHANGE A LOCATION IN THE PROMPRO-7 BUFFER, TYPE "X" AFTER THE "*" PROMPT, FOLLOWED BY A FOUR DIGIT HEX ADDRESS:

```
*X 0001
```

PROMPRO-7 WILL RESPOND

0001 =XX (XX IS A TWO DIGIT HEX VALUE REPRESENTING THE CONTENTS OF LOCATION 0001).

```
TYPE NEW DATA OR, SPACE TO SKIP;
0001 =XX [SPACEBAR]
0002 =XX [55] (AS AN EXAMPLE)
0003 =XX [RETURN]  RETURN TO PROMPT
*
```

NOW IF YOU GO BACK AND EXAMINE LOCATION 0002 YOU SHOULD HAVE THE DATA VALUE "55".

ONCE YOU HAVE COMPLETED YOUR EDITING OF THE RAM BUFFER YOU CAN EXIT THE TERMINAL MODE PROGRAM AND RETURN TO THE CPM OPERATING SYSTEM.

USING PIP COMMAND TO DOWNLOAD

THERE ARE TWO WAYS TO DOWNLOAD A FILE TO THE PROMPRO-7; THE EASIEST METHOD IS APPEND THE PROMPRO-7 ASCII COMMANDS TO THE BEGINNING OF YOUR INTEL HEX FILE USING THE CPM EDITOR:
APPEND THE FOLLOWING STRING OF CHARACTERS TO THE FILE;

```
     ITTTTTTTTTTTTTT
```

IF YOU USE THE TYPE COMMAND TO DISPLAY THE FILE;

```
A>TYPE EZ.HEX        (EZ.HEX IS THE NAME OF THE FILE
                          AS AN EXAMPLE)
```

YOU SHOULD SEE;

```
ITTTTTTTTTTTTT
:100000005C541D.....    (AND THE REST OF YOUR INTEL
                           HEX FILE)
```

NOW USE THE PIP COMMAND TO TRANSFER THE FILE TO THE SERIAL PORT;

A>PIP PUN=A: EZ.HEX (CR)

THIS COPIES THE FILE NAMED EZ.HEX TO THE PUNCH DEVICE. YOU CAN CHANGE THE DEVICE ASSIGNMENT TO WHATEVER YOU WISH USING THE STAT VAL OR SETUP COMMAND OF THE CPM. NOTE; IT IS ASSUMED NO OTHER CHARACTER PRECEDES THE "I" OTHERWISE ENTER THE "I" MANUALLY VIA TERMINAL MODE. REMEMBER THAT RESET CAUSES THE HEX FORMAT BACK TO THE MOTOROLA DEFAULT.

THE SECOND WAY TO DOWNLOAD IS TO USE YOUR EDITOR TO CREATE TWO PERMANENT FILES, EACH CONTAINING ONLY ONE CHARACTER; THE "I" AND THE "T". THEN USING THE SUBMIT FACILITY OF CPM, SEND THE FILE CONTAINING THE "I" , THE "T", AND YOUR HEX FILE WITH ONLY ONE COMMAND. THE ADVANTAGE OF THIS METHOD IS THAT YOU WILL NOT HAVE TO MODIFY YOUR HEX FILE AFTER EVERY ASSEMBLY.

NOTE** SOME 8080 ASSEMBLERS DO NO GENERATE THE PROPER INTEL HEX FORMAT END OF FILE CHARACTER (TEN 00 CHARACTERS ARE GENERATED INSTEAD). THE PROMPRO-7 WILL GENERATE AN ERROR IF THIS LAST LINE OF ZEROS IS ENCOUNTERED. THERE ARE SEVERAL WAYS TO GET AROUND THIS PROBLEM, THE EASIEST WAY IS TO DELETE THE LAST LINE OF THE HEX FILE (TEN 00'S) USING THE EDITOR. PROMPRO-7 WILL TIME OUT (7 SECONDS) AFTER FILE TRANSFER IS COMPLETED. ALSO, YOU MAY TRY USING THE I PARAMETER OF THE PIP COMMAND TO IGNORE THE "00" RECORDS OF HEX FORMAT.

CPM USERS

MANY SOFTWARE DEVELOPMENT TOOLS SUCH AS ASSEMBLERS OR CROSS ASSEMBLERS HAVE AN OPTION THAT ALLOW YOU TO SPECIFY THE HEX ADDRESS DIFFERENTLY THAN THE ACTUAL ASSEMBLY (ORIGIN) ADDRESS. PROMPRO-7 LOADS THE INTEL HEX FILE IN ITS RAM BUFFER AT THE SPECIFIED ADDRESS OF HEX FORMAT. SINCE THE PROGRAMMER ALWAYS PROGRAMS THE EPROM FROM LOCATION 0000, YOU MUST MAKE SURE THAT THE LAST THREE DIGITS OF YOUR HEX ADDRESS IS ALWAYS 000 (PROMPRO-7 IGNORES THE MOST SIGNIFICANT DIGIT). THE XP7 FIRMWARE UPGRADE HAS A FEATURE THAT ALLOWS YOU TO SWAP THE FIRST HALF AND THE SECOND HALF OF THE RAM. THIS IS AN ESPECIALLY USEFUL FEATURE IF YOU CANNOT CHANGE THE HEX ADDRESS.

NOTE; YOU CAN ALSO USE THE PIP COMMAND TO UPLOAD. HOWEVER,IN ORDER TO CLOSE THE FILE, YOU RESET THE UNIT.

TROUBLE - SHOOTING

PROMPRO-7 IS EASY TO OPERATE. HOWEVER IF YOU HAVE A PROBLEM IN OPERATING THE UNIT, FOLLOW THE TROUBLE-SHOOTING GUIDE BELOW:

UPON POWER-UP THERE IS NO LED INDICATION.
CHECK YOUR FUSE HOLDER FOR A BLOWN FUSE.

EPROM DOES NOT VERIFY AFTER LOAD.
IF, AFTER YOU HAVE READ EPROM IN RAM AND DEPRESSED THE VER SWITCH, THE RED LED GOES ON, THE FOLLOWING PROBLEMS MAY EXIST:

* BAD EPROM, REPLACE
* CONFIGURATOR INSERTED BACKWARD, OR HAD BENT PIN.
* EPROM INSERTED BACKWARD.
* RAM NOT PROPERLY IN SOCKET, PRESS IN ALL IC'S.
* LOSS OF VCC, CHECK PIN 24 TO GROUND (PIN 12) FOR 4.75-5.25 VDC WHILE EPROM IN SOCKET. BUSY LED MUST BE ON FOR POWER TO BE APPLIED.

* CHECK OUTPUT ENABLE LINE FOR OE SIGNAL WITH AN OSCILLOSCOPE (THIS IS GENERALLY PIN 20 ON 2716 OR 2732, REFER TO EPROM PIN-OUT APPENDIX D
* CHECK VPP TO BE 5.00 VDC DURING READ.

EPROM DOES NOT VERIFY AFTER PROGRAMMING
IF, AFTER THE PROGRAM CYCLE, THE RED LED GOES ON, IT IS AN INDICATION THAT THE EPROM DID NOT PROGRAM PROPERLY. THE FOLLOWING REASONS COULD ACCOUNT FOR THIS PROGRAM FAILURE:

* EPROM WAS NOT COMPLETELY ERASED.
* EPROM IS DEFECTIVE.
* EPROM IS OF INCORRECT TYPE.
* EPROM INSERTED BACKWARD.
* VPP VOLTAGE FAILURE. CHECK VPP VOLTAGE WHILE IN THE PROGRAM CYCLE.

COMMUNICATIONS PROBLEMS

TERMINAL MODE:

MENU DOES NOT APPEAR ON THE SCREEN AFTER RESET.

CHANCES ARE YOUR CABLING IS INCORRECT. CHECK TO MAKE SURE THAT THE FOLLOWING IS TRUE:

* PROMPRO-7 REAR RS-232 CONNECTOR PIN 3 IS THE TRANSMIT LINE AND SHOULD GO TO THE RECEIVE PIN OF THE TERMINAL WHICH IS ALSO PIN 3 FOR MOST TERMINALS (REFER TO YOUR TERMINAL MANUAL, NOT ALL TERMINALS ARE THE SAME).

* PIN 2 OF PROMPRO-7 IS THE RECEIVE PIN. THIS PIN SHOULD BE CONNECTED TO THE TRANSMIT LINE OF YOUR TERMINAL WHICH SHOULD ALSO BE PIN 2.
* PINS 1 AND 7 ARE GROUND LINES AND SHOULD BE THE ONLY OTHER LINES THAT ARE REQUIRED TO COMPLETE YOUR CONNECTIONS.

YOU MAY BE REQUIRED TO CONNECT PINS 4 TO 5 AND 6 TO 20 ON THE TERMINAL SIDE, IN ORDER TO FORCE THE HANDSHAKE LINES TO THE PROPER LEVELS.

MENU APPEARS ON SCREEN BUT UNIT HANGS-UP.
THIS IS AN INDICATION THAT THE PROMPRO-7 RS-232 HANDSHAKE LINE CTS (PIN 5) IS AT RS-232 LOW SIGNAL LEVEL.
THIS LINE IS INTERNALLY PULLED UP TO +5V.

COMPUTER OPERATION

LED FLICKER ON DOWNLOAD
IF YOU ARE DOWNLOADING A FILE AND YOU SEE THE GREEN AND YELLOW LED FLICKER, THE THE PROMPRO-7 HAS NOT ACCEPTED THE TRANSFER (DOWNLOAD) COMMAND. DURING A SUCCESSFUL DOWNLOAD THE YELLOW (BUSY) LED IS ON SOLID. THIS PROBLEM OCCURS BECAUSE THE USER SENDS THE "T" COMMAND IMMEDIATELY AFTER THE "I". SINCE THE PROMPRO-7 RESPONDS WITH *EOJ* CR LF AFTER IT RECEIVES THE "I" (INTEL FORMAN) COMMAND, IT WILL MISS ANY CHARACTERS SENT TO IT WHILE IT IS BUSY RESPONDING.
ATTEMPT SENDING THE "I" SEPERATELY IN THE TERMINAL MODE.
AFTER THE "T" IS RECEIVED THE PROGRAMMER WILL IGNORE ANY CHARACTER THAT IS NOT A HEADER CHARACTER.
IF THE HEADER CHARACTER IS NOT RECEIVED WITHIN 7 SECONDS, THE PROGRAMMER WILL TIME-OUT.
IF YOU DO NOT WISH TO SEND THE "I" COMMAND MANUALLY IN TERMINAL MODE, YOU MAY SEND IT AUTOMATICALLY IN TWO POSSIBLE WAYS:

* THE FIRST METHOD IS TO CREATE A FILE WITH ONLY THE CHARACTER "I" IN THE FILE, AND ANOTHER FILE WITH ONLY THE CHARACTER "T". NOW YOU MUST USE COMMAND CHAIN CAPABILITY OF YOUR OPERATING SYSTEM TO SEND THE ABOVE FILES AND YOUR HEX FILE TO THE PROGRAMMER WITH ON SIMPLE COMMAND.

* THE SECOND METHOD IS TO ENTER YOUR HEX FILE INTO THE EDITOR AND APPEND THE CHARACTER "T" TO THE BEGINNING OF THE FILE. IF YOU ARE SENDING IT IN THE INTEL HEX FORMAT THEN YOU MUST ALSO APPEND "I" TO THE FILE. TO PROVIDE THE TIMING REQUIRED FOR THE PROMPRO-7 RESPONSE YOU MUST TYPE AT LEAST 15 "T'S" AFTER THE "I".

EXAMPLE:
THIS IS USING THE CPM EDITOR FOR A FILE NAMED PROM.HEX

```
A
ED PROM.HEX
1=*5A APPEND FIVE LINES TO THE TEX BUFFER OF ED EDITOR
1=*ITTTTTTTTTTTTTTT Z    ;TERMINATE INSERT WITH CONTROL Z
 =*E                     ;EXIT EDITOR AND SAVE ON DISK
```

NOW YOUR PROM.HEX FILE HAS THE PROPER COMMAND IN THE BEGINNING OF THE FILE.

TO SEND THE FILE TO THE PROGRAMMER:

```
PIP PUN:=PROM.HEX
```

THIS IS ASSUMING THAT YOU ARE USING THE PUNCH DEVICE PORT FOR YOUR PROM PROGRAMMER.

SINCE USING THE EDITOR TO APPEND THE CONTROL CHARACTERS TO FILE AFTER EVERY ASSEMBLY CAN BECOME QUITE CUMBERSOME, YOU SHOULD TRY TO USE THE SUBMIT COMMAND OF CPM TO SEND THE THREE FILES AUTOMATICALLY:

* I.HEX IS A FILE CONTAINING THE LETTER I ONLY.
* T.HEX IS A FILE CONTAINING THE LETTER T ONLY.
** YOU MUST CREATE THE ABOVE FILES USING THE EDITOR.

* FILE.HEX IS YOUR INTEL HEX FILE GENERATED BY YOUR ASSEMBLER.
* CREATE A FILE (SUBMIT FILE) WITH THE FOLLOWING DATA:
* PIP PUN:=I.HEX,NUL:,T.HEX,$1.HEX
* NAME THE ABOVE FILE "XYZ.SUB"

** NOW TO SEND THE FILE TO THE PROGRAMMER TYPE:
SUBMIT XYZ FILE
WHERE FILE IS THE NAME OF YOUR INTEL HEX FILE.
AFTER A FEW SECONDS OF DISK ACCESS THE PROM-PRO-7 BUSY LIGHT SHOULD GO ON SOLID FOR A FEW SHORT MINUTES.
AFTER A SUCCESSFUL DOWNLOAD, THE GREEN LIGHT SHOULD GO ON.

FOR NON CPM SYSTEMS

THE ABOVE CPM PROCEDURE APPLIES TO MOST SYSTEMS WITH THE DIFFERENCE THAT CERTAIN COMMAND NAMES AND SYSTEMS PARAMETERS MAY VARY. IN ALL CASES, THE IMPORTANT FACT TO REMEMBER IS THAT IF YOUR COMPUTER SENDS ANY CHARACTER OTHER THAN THE PROMPRO-7 COMMANDS, THE PROMPRO-7 WILL RESPOND IN HALF DUPLEX WITH AN "*". SUPPOSE IN SENDING THE CHARACTER "I" YOUR COMPUTER ACTUALLY PRECEDES IT WITH NUL CHARACTER. THEN PROMPRO-7 IS GOING TO MISS THE "I" BECAUSE IT IS BUSY SENDING THE "*". TO GET AROUND THIS PROBLEM, SIMPLY SEND "IIII" INSTEAD OF "I". THIS WILL ALLOW SEVERAL CHARACTER TIME FOR RESPONSE.
IF YOU ENCOUNTER ERROR PROBLEMS DURING TRANSMISSION, CHECK THE GROUND CONNECTION IN YOUR CABLE.
MAKE SURE THAT YOUR SYSTEM IS CONFIGURED FOR:

* ONE START BIT
* 8 DATA BITS
* 2 STOP BITS
* NO PARITY

CALIBRATION CHECK

PROMPRO-7 IS TESTED AND CALIBRATED FOR THE PROPER VOLTAGES PRIOR TO SHIPMENT AT THE FACTORY. THE USER IS NOT REQUIRED TO CALIBRATE THIS UNIT. HOWEVER, SINCE EVEN A SMALL PROGRAM VOLTAGE CHANGE CAN RESULT IN DAMAGE TO AN EPROM, IT IS SUGGESTED THAT USER MAKE PERIODIC CHECKS FOR THE PROGRAM VOLTAGES. PROGRAM VOLTAGE CHANGES CAN HAPPEN ONLY IF CERTAIN COMPONENT FAILURES SURFACE ON THE CNF AND/AN IC IS NOT INSERTED PROPERLY.

THE FOLLOWING TOOLS ARE RECOMMENDED FOR THE VOLTAGE TESTS:

1. 3 1/2 DIGIT DVM
2. 2 SMALL CLIP LEADS
3. 2 TEST PINS

VOLTAGE CHECKS ARE DONE DIRECTLY ON THE SKT1 AND SKT2 SOCKETS. USE TWO TEST PINS OR WIRE LEADS IN THE PROPER PIN LOCATIONS OF THE ZIF SOCKETS. THE DVM CLIP LEADS ARE THEN ATTACHED TO THESE PINS FOR VOLTAGE MEASUREMENT.

ONE OF THE TEST PINS IS ALWAYS USED FOR THE GROUND CONNECTION (PIN 12 ON SKT1 AND PIN 20 ON SKT2). THE SECOND TEST PIN IS INSERTED TO THE CORRESPONDING SOCKET PIN FOR THE PROGRAM VOLTAGE.

25V PROGRAM VOLTAGE CHECK

THIS TEST WILL CHECK THE PROGRAM FOR ALL EPROM'S REQUIRING 25V PROGRAM VOLTAGE:

1. INSERT PIN CONFIGURATOR #1 IN CNF SOCKET.
2. INSERT TEST PINS IN SKT1, PINS 21 AND 12 (CLOSE ZIF SOCKET HANDLE FOR FIRM GRIP).
3. ATTACH DVM GROUND CLIP TO PIN 12 AND THE SECOND CLIP TO PIN 21 (TEST PINS).
4. RESET PROMPRO-7. GREEN L.E.D. SHOULD BE ON. DVM SHOULD READ BELOW 2VDC (SET DVM TO 200VDC SCALE).
5. DEPRESS PROGRAM SWITCH. BUSY L.E.D. SHOULD NOW BE ACTIVATED AND PROGRAM VOLTAGE SHOULD READ BETWEEN 24.3 VDC AND 25.9 VDC.

21VDC PROGRAM CHECK

THIS TEST IS TO CHECK THE VOLTAGE LEVEL FOR EPROM'S REQUIRING 21VDC PROGRAM VOLTAGE (VPP) SUCH AS THE 2732A EPROM.

1. INSERT PIN CONFIGURATOR #4 IN CNF SOCKET.
2. INSERT TEST PINS IN SKT1 PINS 12 AND 20.
3. SET DVM TO 200 SCALE AND ATTACH THE GROUND CLIP TO PIN 12 OF SKT1 AND THE SECOND CLIP TO PIN 20 OF SKT1.
4. RESET PROMPRO-7. DVM READING SHOULD BE BELOW 2VDC.
5. DEPRESS PROG SWITCH ON FRONT PANEL. BUSY L.E.D. WILL INDICATE THAT THE UNIT IS IN THE PROGRAM CYCLE. DVM MUST NOW READ BETWEEN 20.8 VDC AND 21.8 VDC.

8748 (NMOS) VOLTAGE TEST

INSERT PIN CONFIGURATOR #1 IN CNF SOCKET. IN ORDER TO MAKE THE PROPER VOLTAGE TESTS FOR THE 8748 OPERATION, PINS 7, 25, AND 26 OF SKT2 MUST BE TESTED IN THE THREE FOLLOWING STATES:

1. READY STATES: ALL PINS MUST SHOW VOLTAGE LEVELS BELOW 2VDC.
2. ACTIVE STATE: THIS IS WHEN THE PROGRAMMER BUSY LIGHT IS ON DUE TO READ OR VERIFY. IN THIS MODE, PIN 7 SHOULD INDICATE VOLTAGE LEVELS 22.5VDC TO 23.5VDC. PINS 25 AND 26 MUST SHOW VOLTAGE LEVELS 4.75VDC TO 5.25 VDC. THE EASIEST WAY TO MAINTAIN THE PROGRAMMER IN THE ACTIVE STATE IS TO TYPE THE COMMAND "X" FROM A CRT TERMINAL.
3. PROGRAM STATE: IN THIS MODE THE BUSY L.E.D. IS ON AND THE PROGRAMMER IS IN THE PROGRAM CYCLE AS A RESULT OF EXECUTING THE "P" COMMAND OR DEPRESSING THE PROG SWITCH ON THE FRONT PANEL. IN THIS MODE THE 8748 SINGLE CHIP COMPUTER REQUIRES THE FOLLOWING VOLTAGE CHECKS:

   A. SKT2 PIN 25 REQUIRES 22.5VDC TO 23.5VDC
   B. SKT2 PIN 26 REQUIRES 24.5VDC TO 25.5VDC
   C. SKT2 PIN 7 REQUIRES 22.5VDC TO 23.5VDC

IMPORTANT NOTE: BRIDGE PIN 8 AND PIN 40 OF SKT2 USING A 470 OHM 1/4W RESISTOR. THIS IS TO ACTIVATE THE 87XX SENSE LINE TO THE PROMPRO-7. WHITHOUT THIS RESISTOR, SKT2 SOCKET PROGRAM VOLTAGES WILL NOT BE ACTIVATED.

8749H, 8748H PROGRAMMING

THESE DEVICES ARE FABRICATED USING THE HMOS PROCESS, THEREFORE REQUIRING LOWER PROGRAM VOLTAGES THAN THE NMOS VERSION. USE PIN CONFIGURATOR #5 FOR THE 8749H AND PIN CONFIGURATOR #6 FOR THE 8748H.

FOLLOW THE PREVIOUS PROCEDURE AND TEST FOR THE FOLLOWING VOLTAGES USING CNF #5 AND CNF #6:

1. SKT2 PIN 25 REQUIRES 18.5VDC TO 19.5VDC
2. SKT2 PIN 26 REQUIRES 20.5VDC TO 21.5VDC
3. SKT2 PIN 7 REQUIRES 18.5VDC TO 19.5VDC

PROMPRO-7 INTERNAL CIRCUITRY PROVIDES OVER-CURRENT AND THERMAL SHUT-DOWN CIRCUITRY. DEFECTIVE EPROMS CAN CAUSE EXCESSIVE CURRENT FLOW THROUGH THE PROGRAM SUPPLY. WHEN THE PROGRAMMER IS IN THE THERMAL SHUT DOWN STATE, THE UNIT MUST BE TURNED OFF FOR SEVERAL SECONDS AND TURNED BACK ON TO RESUME NORMAL OPERATION.

DO NOT HESITATE TO SEEK ASSISTANCE SHOULD YOU STILL EXPERIENCE ANY FURTHER PROBLEMS.

LOGICAL DEVICES, INC.
1321 N.W. 65 PLACE
FT. LAUDERDALE, FLORIDA 33309
CUSTOMER SERVICE: (305)974-0975

APPENDIX A

HEX FORMATS

INTEL HEX FORMAT DESCIPTION..... PAGE A-1
MOTOROLA HEX FORMAT..... PAGE A-3
8080 ROUTINE 1 TO GEN INTEL HEX.. PAGE A-6
8080 ROUTINE 2 TO GEN INTEL HEX...PAGE A-8
68XX ROUTINE TO GEN MOT. HEX.... PAGE A-10
RELOCATE INTEL HEX ROUTINE.....PAGE A-12

INTEL LOAD MODULE FORMAT

FORMAT DESIGNATION	No. of ASCII CHARS.	CONTENT DESCRIPTION
Header Character	1	Always a colon ":" (HEX 3A)
Record length	2	A two-digit HEX value specifying the number of data bytes in the record. A record length of 0 indicates the last record in the file.
Starting Address	4	Four HEX digits representing the address of the memory location where the first Data Byte will be located.
Type of Record	2	00=normal data record 01=end of file record
Checksum	2	Two-digit HEX number representing two's complement of the sum, modulo 256 of the preceding eight bit bytes.

EXAMPLE :

```
:1000000027040DA5159327D7D5BF8C043A9A8FA83E

: 10 0000 00 27 04 0D A5 15 93 27 D7 D5 BF 8C 04 3A 9A 8F A8 3E
- -- ---- -- ------------------------------------------------ --
| |  |    |                        |                        |
| |  |    |__Type of Record        Data           Checksum__|
| |  |_______Starting Address
| |__________Record Length
|____________Header
```

AN EXAMPLE OF A 1K BYTE INTEL OBJECT MODULE :

```
:1000000027040DA5159327D7D5BF8C043A9A8FA83E
:10001000A999CF54170A722134790234B961804C7
:1000200035994F14F59ACF088A209A7F0214EF363B
:1000300032E7D99625D5ABBF43B5343514423428C1
:10004000043C65C589DF9AA08A802702A8A9AA8AEC
:100050009023C03A8654050937792B6850AD2629534
:1000600004B67657545D5535FAF99CF9ABF03B3B3
:10007000967 4AB930304967BEB029303F5C59682FB
:1000800064060796A9C5AE54CE47AABC60541BECC3
:10009000 8D54708A080A729EBE4026A004A226A231
:1000A000BE80D5FBC532CD24E607C65F03FE96B2FF
:1000B00064BF030896069A8F99CF54178A080A726C
:1000C000C3243BB6CB26C9248744282406233A54AC
:1000D000872310AB54A527547854B1175412EECD92
:1000E000233A54872754A52301547823EF4412C59B
:1000F00018F896F519C58AA09AEFF802F943B03AB4
:100100009AEF802308938948541799 7F541727C57D
:10011000A98A0814F59ACF08AF8A209A7F54FF1451
:10012000EF3624E7D9961593EF8509B2312306240B
:1001300033231554891FFFE39633939A8FF8B65FE4
:1001400002F93A43103A3A3A8002997F238F3923D1
```

```
:100150005C541D8940997F8980346B8A20963BA826
:10016000A93479AF54FF346B9661938A8018F8965E
:1001700078192304367 7E7D983F802F943803A3AAD
:10018000 8AB080089A4F83C514F59A6F800299B798
:10019000 8A88541B9AF718F8968819F972062488E9
:1001A000BAC8D0DBBCCAD5E089060953067703A0DC
:1001B000B3890609530677 03A4B334BCD5BCF0EC6D
:1001C000BFECC1BC1BECC59334CAD5BCBCECCD9311
:1001D000D5BC29ECD3D5BC2AECD893D5BC11ECDE28
:1001E000D5BC10ECE39323535487BB002331540850
:1001F000EEE623535487233954872303547827 5436
:10020000 7827547823FC44125487231354FA54A5B6
:10021000 54B154785470935419541D238362551665
:10022000 2154231627442383 8AA09AEFF802F94326
:10023000303A800299B79ADF541B8A2018F8962822
:10024000 19F93792282406EC530406864BD505BCD1
:10025000DC16477651D5B515BA01B90034B19734DB
:10026000A8FAF72A49F66C866AA9445F34B1F99373
:10027000230D5489230A4489C5AD47547EFD530F8D
:10028000 03305717D287075687D5A8BA019799FE2A
:10029000 0834A8F69FFAF72A58C68E890144918936
:1002A000 0134A824A85478C5F96A54ADF854FA4426
:1002B000 7814F5BF109A6F0854FA547814EFEFB51C
:1002C000FB3793544BC554D047AE54CE4E93544B4A
:1002D00003D0F2E3975792DBB2DB9303EFF2E30331
:1002E000FAF2E6892093031093C554EF530FA954F3
:1002F000C354FAA89354C39A6F02C52B6B2B93086F
:10030000DFC605892093D555FB3291544BD35396C4
:10031000 B85544B530FC5C60BBB00D503F7962175
:10032009554C354FA03FDAD54E9D5FDC63814F510
:10033006 43414EF54F5ED3254C3C537DBD5740182
:10034009 5B60B93111A0D0A65285829616D0D0A8F
:10035004 96E74656C0D0A4D6F746F726F6C610D30
:10036000 A50726F6772616D0D0A5472616E736626
:10037006 5720D0A55706C6F61640D0A566572697D
:10038006 6790D0A2A000D0A2A454F4A2A0D0A2AC3
:10039000 0544BD33A96918554C3ADC5AB54E9D5BF
:1003A0005 4C354FAC6AA95FDC6E414F564B014EF4C
:1003B0005 4F5EDAE54C3C56BD5740195B691932336
:1003C0002 0548754E9D55470C5230F59A9D554A793
:1003D000233D5487 14F59A6F0854782320548754 8A
:1003E0004B03E0C6FC0313C6BEF999DF54C5AC0944
:1003F000B2C6FC9008DCC6FC233F548714EF64C6E9
:00000001FF
```

MOTOROLA LOAD MODULE FORMAT

FORMAT DESIGNATION	No. of ASCII CHARS.	CONTENT DESCRIPTION
Start of Record	1	Always an "S" (HEX 53)
Type of Record	1	0-header record (sometimes not used) 1-normal data record 9-end of file record
Byte Count	2	A two-digit HEX value specifying the number of data bytes in the record, including the address and the check-sum. Each Byte is represented as two HEX characters.
Address	4	Four HEX digits representing the address of the memory location where the first Data Byte will be located.
Checksum	2	Two-digit HEX number representing one's complement of the sum, modulo 256, of the Byte Count, Address, and the Data Bytes.

EXAMPLE :

S113000027040DA5159327D7D5BF8C043A9A8FA83A (normal data record)

```
S 1 13 0000 27 04 CD A5 15 93 27 D7 D5 BF 8C 04 3A 9A 8F A8 3A
- - -- ---- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
|  |  |  |  |                    |                       |
|  |  |  |  |__Starting Address  Data                    Checksum__|
|  |  |  |_____Byte Count
|  |  |________Type of Record
|  |___________Start of Record (header character)
```

AN EXAMPLE OF A 1K BYTE MOTOROLA OBJECT MODULE :

```
S113000027040DA5159327D7D5BF8C043A9A8FA83A
S1130010A999CF54170A7221347902346B961804C3
S113002035994F14F59ACF088A209A7F0214EF3637
S113003032E7D99625D5ABBF43B5343514423428BD
S1130040043C65C589DF9AA08A802702A8A9AA8AE8
S113005090230C03A865405093792B6850AD2629530
S1130060046B676575455D5535FAF99CF9ABF03B3AF
S1130070967 4AB930304967BBB029303F5C59682F7
S1130080640607 96A9C5AE54CE47AABC60541BECBF
S1130090 8D54708A080A729EBE4026A004A226A22D
S11300A0BE80D5FBC532CD24E607C65F03FE96B2FB
S11300B064BF030896069A8F99CF54178A080A7268
S11300C0C3243BB6CB26C92487442824062 33A54A8
S11300D0872310AB54A527547854B1175412EECD8E
S11300E0233A54872754A523015478 23EF4412C597
S11300F018F896F519C58AA09AEFF802F943B03AB0
S1130100 9AEF80230893894854 17997F541727C579
S113011 0A98A0814F59ACF08AF8A209A7F54FF144D
S1130120EF3624E7D9961593BF8509B23123062407
S1130130332315548 91FFFE39633939A8FF8B65FE0
S113014002F93A43103A3A3A8002997F238F3923CD
S1130150 5C541D8940997F8980346B8A20963BA822
S1130160A93479AF54FF346B9661938A8018F8965A
S1130170781923043677E7D983F802F943803A3AA9
S1130180 8AB080089A4F83C514F59A6F800299B794
S1130190 8A88541B9AF718F8968819F972062488E5
S11301A0BAC8D0DBBCCAD5E0890609530677 03A0D8
S11301B0B3890609530677 03A4B334BCD5BCF0EC69
S11301C0BFECC1BC1BECC59334CAD5BCBCECCD930D
S11301D0D5BC29ECD3D5BC2AECD893D5BC11ECDE24
S11301E0D5BC10ECE3932353 487BB00233154084C
S11301F0EEE623535487233954872303547827 5432
S1130200 7827547823FC4412548723 1354FA54A5B2
S1130210 54B1547854709354195 41D238362551661
S1130220 2154231627442383 8AA09AEFF802F94322
S1130230303A800299B79ADF541B8A2018F896281E
S1130240 19F93792282406EC530406864BD505BCCD
S1130250DC16477651D5B515BA01B90034B19734D7
S1130260A8FAF72A49F66C866AA9445F34B1F9936F
S1130270230D5489230A4489C5AD47547EFD530F89
S1130280033057 17D287075687D5A8BA019799FE26
```

```
S11302900834A8F69FFAF72A58C68E890144918932
S11302A00134A824A85478C5F96A54ADF854FA4422
S11302B07814F5BF109A6F0854FA547814EFEFB518
S11302C0FB3793544BC554D047AE54CE4E93544B46
S11302D003D0F2E3975792DBB2DB9303EFF2E3032D
S11302E0FAF2E689209303109 3C554EF530FA954EF
S11302F0C354FAA89354C39A6F02C52B6B2B93086B
S1130300DFC605892093D555FB3291544BD35396C0
S11303100B85544B530FC5C60BBB00D503F7962171
S11303209554C354FA03FDAD54E9D5FDC63814F50C
S1130330643414EF54F5ED3254C3C537DBD574017E
S11303409 5B60B93111A0D0A6528 5829616D0D0A8B
S1130350496E74656C0D0A4D6F746F726F6C610D2C
S11303600A50726F6772616D0D0A5472616E736622
S1130370657 20D0A55706C6F61640D0A5665726979
S1130380667 90D0A2A000D0A2A454F4A2A0D0A2ABF
S11303900 0544BD33A96918554C3ADC5AB54E9D5BB
S11303A054C354FAC6AA95FDC6B414F564B014EF48
S11303B054F5EDAE54C3C56BD5740195B691932332
S11303C020548754E9D55470C5230F59A9D554A78F
S11303D0233D548714F59A6F0854782320 54875486
S11303E04B03E0C6FC0313C6BEF999DF54C5AC0940
S11303F0B2C6FC9008DCC6FC233F548714EF64C6E5
S9030000FC
```

```
S9030000FC    (end of file record)
```

TO GENERATE INTEL HEX FILE

```
        TITLE   'UNLOAD ver 2.1 - Create HEX file from COM file'
;       (revised 05/20/81)
;
;--->Needs MAC and SEQIO.LIB to assemble<---
;
;Originally from CPMUG 29.23
;
;05/20/81 Modified for 32 bytes/record instead of 16, for less
;         overhead in the .HEX output file, by Dav Holle.
;
;11/07/80 Modified to default to 100H, increase size of buffers,
;         add signon message.  By Keith Petersen, W8SDZ
;
;To use, type: UNLOAD <FILENAME> <ADDR>
;
;Where: <FILENAME>.COM is the source file
;       <FILENAME>.HEX will be the output file
;       <ADDR> is the optional start address in hex (default=100)
;
        ORG     100H
;
        MACLIB  SEQIO   ;DEFINE MACRO LIBRARY USED
;
        LHLD    6       ;GET BASE OF BDOS
        DCX     H       ;BACK OFF ONE BYTE
        SPHL            ;SET STACK THERE
        CALL    SIGNON
        DB      CR,LF,'UNLOAD ver 2.1',CR,LF,'$'
        DB      '05/20/81' ;REVISION DATE (doesn't print)
;
SIGNON: POP     D       ;GET MSG ADR
        MVI     C,@MSG  ;PRINT IT
        CALL    @BDOS
        LXI     H,100H  ;DEFAULT UNLOAD ADRS
        LXI     D,FCB2+1
        LDAX    D       ;GET OPTION ADRS
        CPI     ' '     ;ANY GIVEN?
        JZ      INITFL  ;NO, DEFAULT TO 100H
        LXI     H,0
        MVI     B,0
;
ADRLUP: LDAX    D
        INX     D
        SUI     '0'
        JC      INITFL
        CPI     10
        JC      ADDNIB
        SUI     7
        JC      INITFL
        CPI     16
        JNC     INITFL
;
ADDNIB: DAD     H
```

```
        DAD     H
        DAD     H
        MOV     C,A
        DAD     B
        JMP     ADRLUP
;
INITFL: PUSH    H
        FILE    INFILE.SOURCE,,1,COM,2048
        FILE    OUTFILE,OUTPUT,,1,HEX,2048
        POP     H
;
ADRDON: SHLD    LODADR
;
UNLOOP: GET     SOURCE
        JZ      GEOF
        PUSH    PSW
        MVI     A,':'
        PUT     OUTPUT
        XRA     A
        STA     CHEKS
        MVI     A,32    ;was 16
        CALL    PUTBYTE
        LDA     LODADR+1
        CALL    PUTBYTE
        LDA     LODADR
        CALL    PUTBYTE
        XRA     A
        CALL    PUTBYTE
        POP     PSW
        MVI     B,32    ;was 16
;
LINLUP: PUSH    B
        CALL    PUTBYTE
        POP     B
        DCR     B
        JZ      NEXTL
        GET     SOURCE
        JMP     LINLUP
;
PUTBYTE:MOV     C,A
        LDA     CHEKS
        SUB     C
        STA     CHEKS
        MOV     A,C
        RRC
        RRC
        RRC
        RRC
        CALL    PUTNIB
        MOV     A,C
;
PUTNIB: ANI     0FH
        ADI     '0'
        CPI     '9'+1
        JC      PUTNB1
        ADI     7
;
PUTNB1: PUSH    B
        PUT     OUTPUT
        POP     B
        RET
;
NEXTL:  LDA     CHEKS
        CALL    PUTBYTE
        MVI     A,CR
```

MVI A,CR

```
        PUT     OUTPUT
        MVI     A,LF
        PUT     OUTPUT
        LHLD    L' DADR
        LXI     D,32    ;was 16
        DAD     D
        JMP     ADRDON
;
GEOF:   MVI     A,':'
        PUT     OUTPUT
        MVI     B,5
;
GEOF1:  XRA     A
        PUSH    B
        CALL    PUTBYTE
        POP     B
        DCR     B
        JNZ     GEOF1
        MVI     A,CR
        PUT     OUTPUT
        MVI     A,LF
        PUT     OUTPUT
        FINIS   OUTPUT
        LXI     D,DMSG
        MVI     C,@MSG
        CALL    @BDOS
        JMP     0
;
DMSG:   DB      'DONE',CR,LF,'$'
;
LODADR: DS      2
CHEKS:  DS      1
FCB2    EQU     6CH
;
BUFFERS EQU     $       ;INPUT/OUTPUT BUFFERS GO HERE
;
        END
```

8080 Program to Generate Intel Hex Format

```
        READ COMMAND   READ INTEL HEX FORMAT TAPE (W/BIAS IF DESIRED)
READ:   CALL   EXPRI   ;GET BIAS INPUT IF DESIRED
        POP      H     ;PLACE BIAS INTO HL (IS 0000 OF NO BIAS)
REDO:   CALL    RI     ;READ FROM LOGICAL TAPE READER DEVICE
        RC             ;RETURN IF NO BYTE IS AVAILABLE
        ANI    7FH     ;MASK OFF POSSIBLE BITS
        SUJ    ':'     ;CLUMSY COMPARISON FOR START LINE
        JNZ    REDO    ;LOOP TIL START OF LINE DELIMINATOR FOUND
        MOV     D,A    ;A MUST BE ZERO, SO USE IT TO CLEAR D RFG
        PUSH     H     ;SAVE BIAS VALUE ON STACK
        CALL   BYTE    ;GET FIRST DATA BYTE
        JZ     RED2    ;IF ZERO, INDICATES LAST LINE AND LOAD ADDRESS
        MOV     F,A    ;LOAD NUMBER OF DATA BYTES IN LINE INDICATOR IN
        CALL   BYTE    ;GET FIRST ADDRESS BYTE
        MOV     B,A    ;SAVE HIGH ORDER ADDRESS IN B
        CALL   BYTE    ;GET LOWER ADDRESS BYTE
        MOV     C,A    ;SAVE LOW ORDER ADDRESS IN C
        DAD      B     ;ADD BIAS TO LOADING ADDRESS FROM LINE
        CALL   BYTE    ;JUST READ TERMINATOR BYTE
RED1:   CALL   BYTE    ;READ DATA BYTE
        MOV     M,A    ;STORE DATA BYTE AT LOAD ADDR+BIAS
        INX      H     ;BUMP STORAGE POINTER
        DCR      F     ;DECREMENT BYTE COUNT FOR LINE
        JNZ    RED1    ;LOOP TILL BYTE COUNT IS EXHAUSTED
        CALL   BYTE    ;READ CHECK BYTE AND DO FINAL SUM
        POP      H     ;RESTORE BIAS FOR NEXT LINE
        JZ     REDO    ;JUMP FOR NEXT LINE IF CHECKSUM=0
        STC            ;SET THE CARRY BIT ON AS AN ERROR INDICATOR
        RFT
RED2:   CALL   BYTE    ;GET TWO ADDRESS BYTES AND BRANCH THERE
        MOV     H,A
        CALL   BYTE
        POP      B     ;REMOVE BIAS VALUE FROM STACK
        MOV     L,A    ;CHECK THE EXECUTION ADDRESS FOR ZERO
        URA      H
        RZ             ;RETURN TO MONITOR ON A ZERO EXECUTION ADDR
        PCH1           ;GO TO EXECUTION ADDRESS
BYTE:   CALL  RNBBL    ;READ UPPER NIBBLE OF DATA BYTE
        RLC            ;SHIFT TO UPPER 4 BITS
        RLC
        RLC
        RLC
        MOV     C,A    ;SAVE MS NIBBLE IN C
        CALL  RNBBL    ;READ SECOND NIBBLE
        ORA      C     ;COMBINE NIBBLES TO FORM HEX BYTE
        MOV     C,A    ;SAVE BYTE IN C
        ADD      D     ;ADD TO CHECK SUM
        MOV     D,A    ;SAVE NEW CHECK SUM
        MOV     A,C    ;RESTORE DATA BYTE TO ACCUM
```

```
RNBB1:  CALL     RJ     ;DO READER INPUT
        JC     RNBER    ;IF CARRY SET, ERROR
        ANJ      ZFH    ;MASK OFF PARITY BIT
        CALL   NIBBL    ;READ NIBBLE FROM ASCII HEX BYTE
        JC      RNBR    ;JUMP ON ERROR
        RET
RNBFR:  POP      H      ;RETURN TO MONITOR ON ERROR
        POP H
        POP H
        RET
```

```
FC11 17  01A1  LOAD1   LBSR    ECHON    INBUT 8 BIT BYTE WITH NO ECHO
FC14 81  53    LOAD2   CMPA    #'S      IS IT S, START CHAR?
FC16 26  F9            BNE     LOAD1    NO, DISCARD AND GET NEXT CHAR
FC18 17  019A          LBSR    ECHQN
FC1B 81  39            CMPA    #'9      IS 9, END OF FILE CHAR?
FC1D 27  3D            BEQ     LOAD21   IS SO, EXIT LOAD
FC1F 81  31            CMPA    #'1      IS 1, FILE LOAD CHAR?
FC21 26  F1            BNE     LOAD2    IF NOT, LOOK FOR START CHAR
FC23 17  0116          LBSR    BYTE     INPUT BYTE CNT
FC26 34  02            PSHS    A        PUSH COUNT ON STK
FC28 29  26            BVS     LODERR   (V)C-CODE SET, ILLEGAL HEX
FC2A 17  00FE          LBSR    IN1ADR   INPUT LOAD ADDR
FC2D 29  21            BVS     LODERR   (V) C-CODE SET, ADDR NOT HEX
FC2F 34  10            PSHS    X        PUSH ADDR ON STK
FC31 E6  E0            LDB     ,S+      LOAD MSB OF ADDR AS CKSUM BYTE
FC33 EB  E0            ADDB    ,S+      ADD LSB OF ADDR TO CKSUM
FC35 EB  E4            ADDB    ,S       ADD BYTE COUNT BYTE TO CKSUM
FC37 6A  E4            DEC     ,S       $FC37 DEC BYTE COUNT 2 TO BYPASS
FC39 6A  E4            DEC     ,S       ADDRESS BYTES
FC3B 34  04    LOAD10  PSHS    B        PUSH CKSUM ON STK
FC3D 17  00FC          LBSR    BYTE     INPUT DATA BYTE - 2 HEX CHARS
FC40 35  04            PULS    B        POP CKSUM FROM STK
FC42 29  0C            BVS     LODERR   (V) SET, DATA BYTE NOT HEX
FC44 34  02            PSHS    A        PUSH DATA BYTE ON STK
FC46 EB  E0            ADDB    ,S+      ADD DATA TO CKSUM, AUTO INC STK
FC48 6A  E4            DEC     ,S       DEC BYTE COUNT 1
FC4A 27  05            BEQ     LOAD16   IF BYTE COUNT ZERO, CHECK CKSUM
FC4C A7  80            STA     ,X+      SAVE DATA BYTE IN MEM
FC4E 20  EB            BRA     LOAD10   GET NEXT DATA BYTE
FC50 5F        LODERR  CLRB             ERR COND, ZERO CKSUM
FC51 35  02    LOAD16  PULS    A        ADJUST STK (REMOVE BYTE CNT)
FC53 C1  FF            CMPB    #$FF     CKSUM OK?
FC55 27  B2            BEQ     LOAD     IF SO, LOAD NEXT LINE
FC57 86  3F            RDA     #'?      LOAD "?", ERR INDICATOR
FC59 17  0183          LBSR    OUTCH    OUTPUT TO TERM
FC5C 73  DFE2  LOAD21  COM     ECHO     TURN ECHO ON
FC5F 86  13            LDA     #$13     $FC5F LOAD "DC3" CASS READ OFF COD
FC61 16  017B          LBRA    OUTCH    OUTPUT IT
                       *
                       * "P" PUNCH MIKBUG TAPE
                       *
FC64 6F  E2    PUNCH   CLR     ,-S      CLEAR RES BYTE ON STK
FC66 17  00B7          LBSR    IN2ADR   GET BEG & END ADDR
FC69 34  30            PSHS    X,Y      SAVE ADDR'S ON STK
FC6B 29  4A            BVS     PUNEXT   (V) C-CODE SET, EXIT PUNCH
FC6D AC  62            CMPX    2,S      CMP BEG TO END ADDR
FC6F 25  46            BCS     PUNEXT   IF BEG>END, EXIT PUNCH
FC71 30  01            LEAX    1,X      INC END ADDR
FC73 AF  E4            STX     ,S       STORE END ADDR ON STK
FC75 86  12            LDA     #$12     LOAD 'DC2' PUNCH ON CODE
```

```
FC77 17   0165          LBSR   OUTCH    OUTPUT TO TERM
FC7A EC   E4    PUNCH2  LDD    ,S       LOAD END ADDR IN D REG
FC7C A3   62            SUBD   2,S      SUB BEG FROM END
FC7E 27   06            BEQ    PUNCH3   SAME, PUNCH 32 BYTES OF DEFAULT
FC80 1083 0020          CMPD   #$20     LESS THAN 32 BYTES?
FC84 23   02            BLS    PUNCH4   PUNCH THAT MANY BYTES
FC86 C6   20    PUNCH3  LDB    #$20     LOAD BYTE CNT OF 32
FC88 E7   64    PUNCH4  STB    4,S      STORE ON STK AS BYTE CNT
FC8A 8E   FEDC          LDX    #MSG20   POINT TO MSG "S1"
FC8D 17   010E          LSBR   PSTRNG   PRINT MSG
FC90 CB   03            ADDB   #3       ADD 3 BYTES TO BYTE CNT
FC92 1F   98            TFR    B,A      GET BYTE CNT I A-REG TO PUNCH
FC94 17   00DB          LBSR   OUT2H    OUTPUT BYTE COUNT
FC97 AE   62            LDX    2,S      LOAD BEG ADDR
FC99 17   00CE          LBSR   OUT4H    PUNCH ADDR
FC9C EB   62            ADDB   2,S      ADD ADDR MSB TO CKSUM
FC9E EB   63            ADDB   3,S      ADD ADDR LSB TO CKSUM
FCA0 EB   84    PUNCHL  ADDB   ,X       ADD DATA BYTE TO CKSUM
FCA2 A6   80            LDA    ,X+      LOAD DATA BYTE TO PUNCH
FCA4 17   00CB          LBSR   OUT2H    OUTPUT DATA BYTE
FCA7 6A   64            DEC    4,S      DEC BYTE CNT
FCA9 26   F5            BNE    PUNCHL   NOT DONE, PUNCH NEXT BYTE
FCAB 53                 COMB            1'S COMPLIMENT CKSUM BYTE
FCAC 1F   98            TFR    B,A      PUT IN A REG TO PUNCH
FCAE 17   00C1          LBSR   OUT2H    OUTPUT CKSUM BYTE
FCB1 AF   62            STX    2,S      SAVE X REG IN STK AS NEW PUNCH ADD
FCB3 AC   E4            CMPX   ,S       COMPARE TO END ADDR
FCB5 26   C3            BNE    PUNCH2   $FCB5 PUNCH NOT DONE, CONT
FCB7 86   14    PUNEXT  LDA    #$14     LOAD "DC4" PUNCH OFF CODE
FCB9 17   0123          LBSR   OUTCH    OUTPUT IT
FCBC 32   65            LEAS   5,S      READJ STK PTR
FCBE 39                 RTS
```

```
10 'PROGRAM NAME IS                "RELOC"
20 '
30 'THIS PROGRAM RELOCATES THE CODE IN AN EXISTING INTEL HEX FILE NAMED
40 ' "FILENAME.HEX". IT STARTS THE CODE AT THE DESIRED STARTING ADDRESS,
50 'RECALCULATES THE CHECKSUMS AND OUTPUTS THE MODIFIED FILE WITH THE NAME
60 ' "FILENAME.REL".  THE PROGRAM ALSO KEEPS A RUNNING TOTAL OF
70 'THE NUMBER OF BYTES OF CODE THAT ARE BEING TRANSFERRED TO THE
80 'NEW FILE.
90 INPUT"NAME OF HEX FILE TO RELOCATE";NA$
100 INPUT"DESIRED STARTING ADDRESS";SA$
110 SA=VAL("&H"+SA$):CB=0
120 PRINT
130 PRINT"NUMBER OF BYTES WRITTEN FOLLOWS"
140 NAMIN$=NA$+".HEX"
150 OPEN"I",#1,NAMIN$
160 NAMOUT$=NA$+".REL"
170 OPEN"O",#2,NAMOUT$
180 IF EOF(1) THEN 400:'           STOP ON END OF FILE
190 LINE INPUT#1,A$:'              GET RECORD
200 X$=MID$(A$,2,2)
210 NTOT=VAL("&H"+X$):'           NUMBER OF BYTES IN RECORD
220 IF NTOT<>0 THEN 250:'         CHECK IF LAST RECORD
230 PRINT#2,A$:'                  OUTPUT LAST RECORD
240 GOTO 400:'                    STOP
250 CS=10+2*NTOT:'                 LOCATION OF CHECKSUM CHARACTERS
260 NC=LEN(SA$):'                  NUMBER OF CHARACTERS IN ADDRESS
270 IF NC=4 THEN SAA$=SA$
280 IF NC=3 THEN SAA$="0"+SA$
290 IF NC=2 THEN SAA$="00"+SA$
300 IF NC=1 THEN SAA$="000"+SA$
310 MID$(A$,4,4)=SAA$:'            SET NEW STARTING ADDRESS
320 GOSUB 440:'                CALL CHECKSUM
330 MID$(A$,CS,2)=CS$:'            SET NEW CHECKSUM
340 PRINT#2,A$:'                  OUTPUT MODIFIED RECORD
350 SA=SA+NTOT:'                  INCREMENT STARTING ADDRESS
360 CB=CB+NTOT:'                 CUMULATIVE NUMBER OF BYTES
370 PRINT CB
380 SA$=HEX$(SA)
390 GOTO 180:'                     DO IT AGAIN
400 CLOSE#1
410 CLOSE#2
420 PRINT"MODIFIED FILE IS CALLED,",NAMOUT$
430 STOP
440 '.................SUBROUTINE CHECKSUM.........................
450 SUM=0:X=0
460 FOR J=2 TO 8+2*NTOT  STEP 2
470 N$=MID$(A$,J,2)
480 GOSUB 560:'              CALL HEXTODEC
490 SUM=SUM+X
500 IF SUM>256 THEN SUM=SUM-256
510 NEXT J
520 CS$=HEX$(256-SUM)
530 LC=LEN(CS$)
540 IF LC=1 THEN CS$="0"+CS$
550 RETURN
560 '.................SUBROUTINE HEXTODEC.........................
570 X=VAL("&H"+N$)
580 IF X<0 THEN X=X+2^16
590 RETURN
```

relocated hex file

FILE.REL

```
:08[0000]00E6F6003130ECFFFCD4
:08000800FDD61062D61057D698
:0800100010624C10D610694A81
:0800180FBD610E2D610AC8DFE
:0800200103E7C122C00923015E
:0800280A0E07AFAAF31300CC0
:0800300181C00AFE602306C61
:0800380026AFEE602106C0FE3
:08004000EAFE3C00D610627C30
:0800480006D610847AFBAF59C3
:0800500002822E8230023928D
:0800580205E5902822E56E2DF
:0800600007FA0E0823012239220
:0800680020A0E0FD10A8823089
:0800700E3E9230A0E05EAF7C7F
:0800780006D61062828A0E0B0
:0800800082290A0E082A0BC0404
:0800880FD10C1EFC0EAC0E960
:0800900C0E8BAF499F0D610A3
:0800980D889F0D610D8A0E0D1
:0800A0007ADAAF76FA106D1058
:0800A800D8E6FA00AFBC0CE63B
:0800B000F0FFD610D8BAF8E603
:0600B800F000D610D8AFE5
:00000001FF
```

starts 0000H

original hex file

FILE.HEX

starts 1035 H

```
:0810350000E6F6003130ECFFFC8F
:08103D00FDD61062D61057D653
:0810450010624C10D610694A3C
:08104D00FBD610E2D510AC8DB9
:0810550010 3E7C122C009220D9
:08105D00A0E07AFAAF31300C7B
:0810650018 1C00AFE602306C1C
:08106D00026AFEE602106C0F9E
:0810750006AFE5C00D610627CEB
:08107D0006D610847AF8AF597E
:0810850002822E8230022392 48
:08108D00205E5902822E56E29A
:0810950007FA0E0823012239 2DB
:08109D0020A0E0FD10A8823044
:0810A5003E9230A0E05EAF7C3A
:0810AD0006D6106282 80A0E06B
:0810B5008290A0E092A0BC04BF
:0810BD00FD10C1EFC0EAC0E91B
:0810C500C0E8BAF499F0D6105E
:0810CD00D889F0D610D8A0E08C
:0810D5007ADAAF76FA106D1013
:0810DD00D8E6FA00AFBC0CE6F6
:0810E500F0FFD610D8BAF8E68E
:0610ED00F000D610D8AFA0
:00000001FF
```

APPENDIX B

SOFTWARE DRIVERS

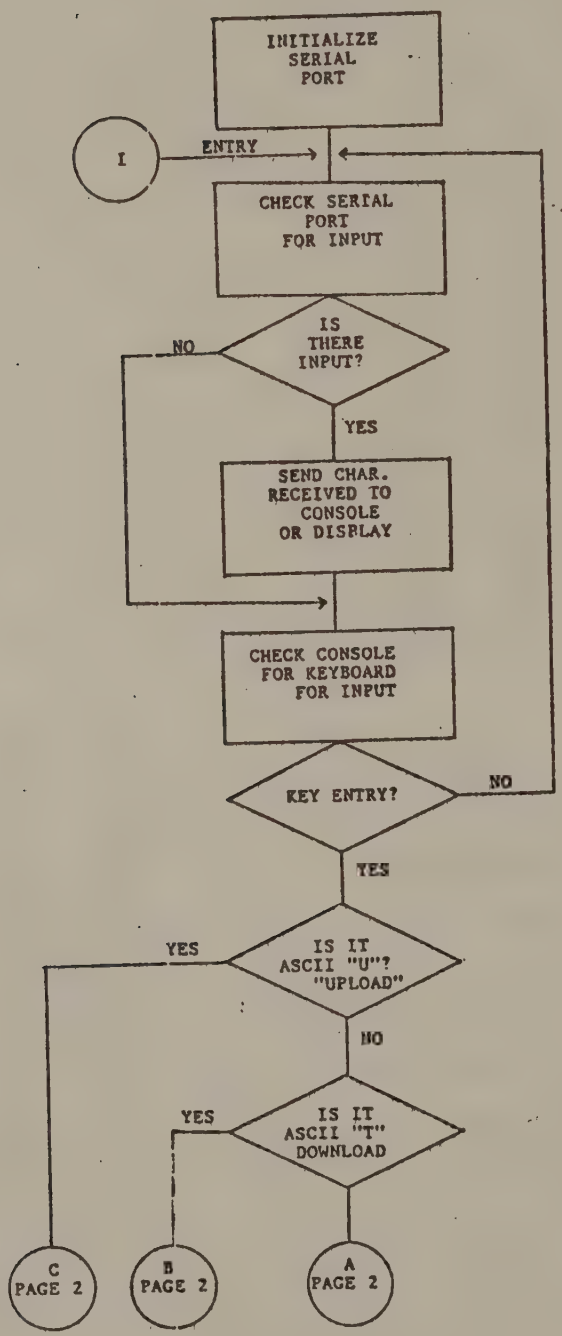
TABLE OF STANDARD ASCII CODES.....PAGE B-1
GENERALIZED SOFTWARE DRIVER FLOW-CHART.....PAGE B-2
COMMODORE-64 SOFTWARE DRIVER.....PAGE B-4
TRS-80 COLOR SOFTWAER DRIVER.....PAGE B-10
SWTPC/FLEX SOFTWARE DRIVER.....PAGE B-13
TEKTRONICS 8002.....PAGE B-14

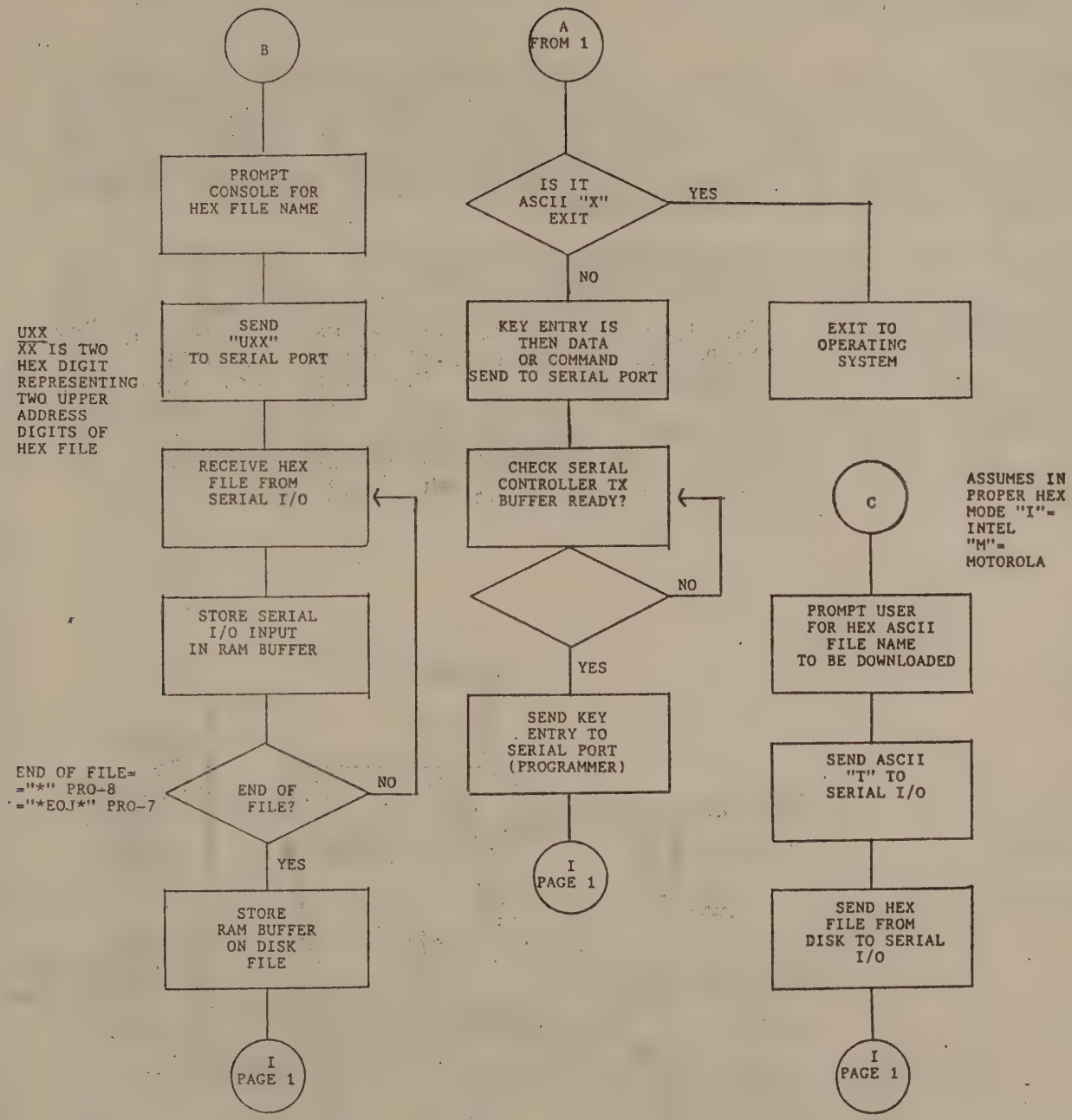
ASCII CODES (HEXIDECIMAL)

				B7	0	0	0	0	0	0	0	0
				B6	0	0	0	0	1	1	1	1
				B5	0	0	1	1	0	0	1	1
				B4	0	1	0	1	0	1	0	1
BINARY				HEX								
B3	B2	B1	B0	$	0	1	2	3	4	5	6	7
0	0	0	0	0	NUL	DLE	SP	0	@	P	\`	p
0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	2	STK	DC2	"	2	B	R	b	r
0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
0	1	0	0	4	EOT	DC4	$	4	D	T	d	t
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
0	1	1	1	7	BEL	ETB	\	7	G	W	g	w
1	0	0	0	8	BS	CAN	(	8	H	X	h	x
1	0	0	1	9	HT	EM	)	9	I	Y	i	y
1	0	1	0	A	LF	SUB	*	:	J	Z	j	z
1	0	1	1	B	VT	ESC	+	;	K	[	k	{
1	1	0	0	C	FF	FS	,	<	L	\	l	!
1	1	0	1	D	CR	GS	-	=	M	]	m	}
1	1	1	0	E	SO	RS	.	>	N	^	n	~
1	1	1	1	F	S1	US	/	?	O	_	o	RUBOUT

NUL	Null	DLE	Data Link Escape
SOH	Start of Heading	DC1	Device Control 1
STX	Start of Text	DC2	Device Control 2
ETX	End of Text	DC3	Device Control 3
EOT	End of Transmission	DC4	Device Control 4
ENQ	Enquiry	NAK	Negative Acknowledge
ACK	Acknowledge	SYN	Synchronous Idle
BEL	Bell	ETB	End of Transmission
BS	Back Space	CAN	Cancel
HT	Horizontal Tabulation	EM	End of Memory
LF	Line Feed	SUB	Substitute
VT	Vertical Tabulation	ESC	Escape
FF	Form Feed	FS	File Separator
CR	Carriage Return	GS	Group Separator
SO	Shift Out	RS	Record Separator
SI	Shift In	US	Unit Separator

GENERALIZED SOFTWARE DRIVER FLOW CHART



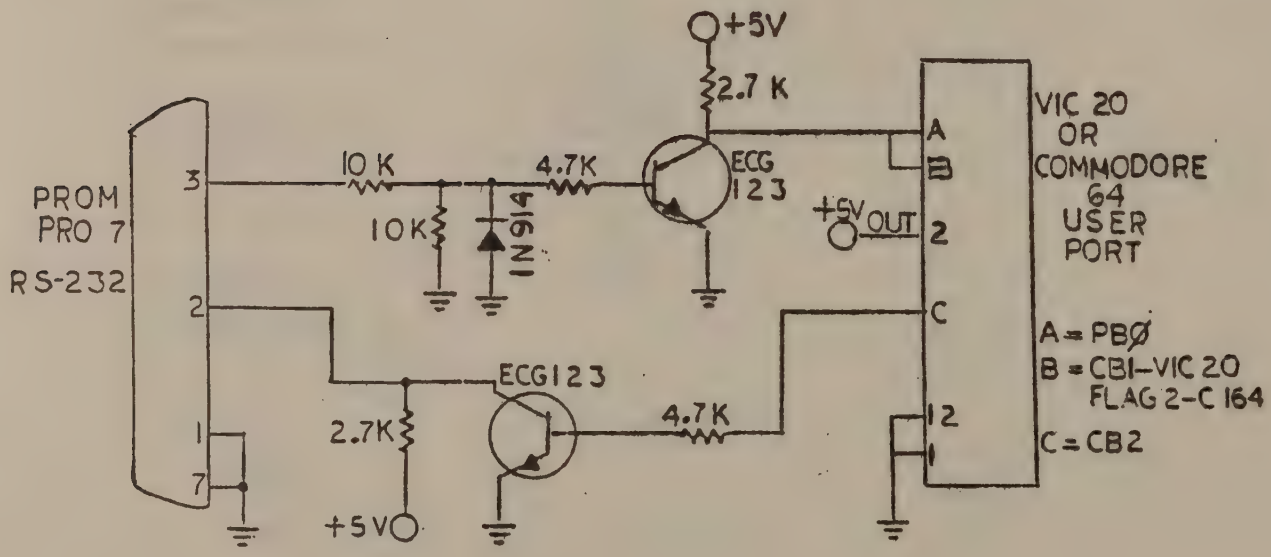

B
PROMPT CONSOLE FOR HEX FILE NAME
SEND "UXX" TO SERIAL PORT
UXX
XX IS TWO HEX DIGIT REPRESENTING TWO UPPER ADDRESS DIGITS OF HEX FILE
RECEIVE HEX FILE FROM SERIAL I/O
STORE SERIAL I/O INPUT IN RAM BUFFER
END OF FILE?
NO
YES
END OF FILE= ="*" PRO-8 ="*EOJ*" PRO-7
STORE RAM BUFFER ON DISK FILE
I PAGE 1
A FROM 1
IS IT ASCII "X" EXIT
YES
NO
KEY ENTRY IS THEN DATA OR COMMAND SEND TO SERIAL PORT
EXIT TO OPERATING SYSTEM
CHECK SERIAL CONTROLLER TX BUFFER READY?
NO
YES
SEND KEY ENTRY TO SERIAL PORT (PROGRAMMER)
I PAGE 1
C
ASSUMES IN PROPER HEX MODE "I"= INTEL "M"= MOTOROLA
PROMPT USER FOR HEX ASCII FILE NAME TO BE DOWNLOADED
SEND ASCII "T" TO SERIAL I/O
SEND HEX FILE FROM DISK TO SERIAL I/O
I PAGE 1

COMMODORE 64 INSTRUCTIONS

THE FOLLOWING IS A LISTING OF THE MODIFIED SOFTWARE DRIVER FOR THE COMMODORE 64 COMPUTER.

WHEN THE PROGRAM IS RUN, A MENU IS DISPLAYED AND THE OPERATOR IS ASKED TO CHOOSE BETWEEN CLEARING THE C-64 BUFFER, OR ENTERING DATA. IF THE USER WISHES TO CLEAR THE BUFFER, $00 IS PLACED IN ALL LOCATIONS FROM $C000 TO $C400 CREATING A CLEARED BUFFER AREA 1K WIDE. BY CHANGING THE NUMBER 1024 TO ANY OTHER NUMBER UP TO 4069, A LARGER BUFFER AREA MAY BE CLEARED.

IF THE USER WANTS TO ENTER DATA INTO THE BUFFER, THE PROGRAM WILL PRINT A 3-DIGIT HEXADECIMAL ADDRESS (STARTING AT $000) AND THE USER CAN ENTER A 2-DIGIT HEXADECIMAL VALUE TO BE STORED AT THAT ADDRESS IN PROMPRO MEMORY. THE USER MAY USE THE DELETE KEY TO CORRECT ANY ERRORS AND THE COMPUTER WILL NOT ACCEPT ANY CHARACTERS OTHER THAN THE DIGITS 0 - 0 AND THE LETTERS A - F. THE ENTER KEY MUST BE PRESSED AFTER EACH ENTRY. THIS SCHEME SEEMS TO REDUCE ERRORS TO A MINIMUM. AFTER THE LAST ENTRY, THE USER PRESSES THE "Q" KEY TO QUIT THE DATA ENTRY PROGRAM. THE PROGRAM AUTOMATICALLY ASSIGNS THE BEGINNING AND END ADDRESSES TO THE UPPER AND LOWER LIMITS.



```
1       OPEN5,2,3,CHR$(6+128)+CHR$(16):SR=ST
2       GOTO2000
3       REM***SOFTWARE DRIVER FOR PROMPRO7 TORUN ON LENOMMODORE-64 COMPUTER
4       REM***600 BAUD
6       DIMA%(75),T%(150),C%(255),E%(255),F%(70)
7       FORJ=32TO64:T%(J)=J:NEXT
8       T%(13)=13:T%(20)=8:RV=18:CT=0
9       FORJ=65TO90:T%(J)=J:NEXT
10      FORJ=91TO95:T%(J)=J:NEXT
11      FORJ=97TO122:K=J-32:T%(J)=K:NEXT
12      T%(146)=16:T%(133)=16
18      FORJ=65TO70:K=J-55:A%(J)=K:NEXT
22      FORJ=0TO9:K=J+48:A%(J)=K:NEXT
23      FORJ=48TO57:K=J-48:A%(J)=K:NEXT
24      FORJ=10TO16:K=J+55:A%(J)=K:NEXT:GOSUB625
25      POKE828,83:POKE829,49:T=15
26      POKE830,49:POKE831,51:C2=836:GOSUB700
27      PRINTCHR$(147);
28      FORVV=49152TOBA
29      VW=PEEK(VV):PRINTVW,:NEXTVV
31      PRINTCHR$(13);CHR$(13);
35      H=49152:H5=H
40      Y=BA:Y5=Y
50      J2=5:F=H:GOSUB05
51      G$=CHR$(A%(W)):GOSUB893:H=F:J2=0
52      GOTO210
55      GOSUB800
60      J2=5:H=N1+19+A5+Z1
65      GOSUB805:H=F:N1=0
85      GOSUB985
89      GOSUB893:J2=0
90      PRINT
97      IFF>YTHENGOSUB750:GOTO210
98      C2=836
99      GOTO55
200     REM KEYBOARD
210     GET#5,A$
220     IFA$=""ORST<>0THEN260
230     PRINT" "CHR$(157);CHR$(T%(ASC(A$)));
240     IFT%(ASC(A$))=34THENPOKE8,0
250     GOTO210
260     PRINTCHR$(RV)" "CHR$(157);CHR$(146);:GETA$
267     IFT7=1THEN375
270     IFA$<>""THENPRINT#5,CHR$(T%(ASC(A$)));:PRINTA$;
280     CT=CT+1
290     IFCT=8THENCT=0:RV=164-RV
291     IFM7=1THENM7=0:GOTO1000
292     IFA$="T"THENPRINTCHR$(13);:GOTO55
293     IFA$="U"THENA$=G$:PRINTCHR$(13);:M7=1:GOTO2500
294     IFA$="C"THEN2
295     IFA$="B"GOTO2000
296     GOTO210
```

```
299   REM PRINTS "S113
350   FORI=828TO831
351   I3=PEEK(I)
352   A$=CHR$(I3)
353   PRINT#5,CHR$(ASC(A$));:PRINTA$;
357   CT=CT+1
360   IFCT=8THENCT=0:RV=164-RV
365   NEXTI
370   RETURN
600   PRINT#5,CHR$(ASC(A$));:PRINTA$;
605   RETURN
625   C=0:D=9:E=48:F=57
630   G=55:H=60:L=10:M=15
635   FORI=1TO16
637   FORB=CTOD:K=B+E:C%(B)=K:NEXTB
640   FORJ=LTOM:K=J+G:C%(J)=K:NEXT
643   C=C+16:D=D+16:L=L+16:M=M+16:E=48-C:G=65-L
650   NEXTI
655   C1=0:D1=15:R=48
660   FORX=1TO16
667   FORQ=C1TOD1:E%(Q)=R:NEXTQ:IFR=57THENR=64
670   C1=C1+16:D1=D1+16:R=R+1
672   NEXTX
675   K=0
676   FORJ=48TO57
677   F%(J)=K:K=K+16:NEXTJ
680   FORJ=65TO70:F%(J)=K:K=K+16:NEXTJ
685   RETURN
695   NEXTY4
699   REM END OF FILE
700   POKE870,83:POKE871,57
701   POKE872,48:POKE873,51
702   POKE874,48:POKE875,48
703   POKE876,48:POKE877,48
704   POKE878,70:POKE879,67
705   RETURN
750   FORI=870TO879
755   I4=PEEK(I)
757   A$=CHR$(I4)
760   PRINT#5,CHR$(ASC(A$));:PRINTA$;
765   CT=CT+1
770   IFCT=8THENCT=0:RV=164-RV
775   NEXTI
780   RETURN
785   REM DEC. TO HEX TALLY VAL OF MEMORY
800   N1=0
803   F=H
804   N=PEEK(H)
805   IFH>4096THENH=H-4096:W=W+1:GOTO805
810   IFH<4096THENH1=H
855   IFH1>=256THENH1=H1-256:W1=W1+1;GOTO855
860   IFH1<256THENH2=H1
```

```
1080  RETURN
1900  REM***ALLOWS E'TRY OF HEX DATA WHICH IS STORED IN BINARY OBJECT FORM
1910  REM***IN A BUFFER BEGINNING AT 49152 ($C000)
1920  REM***CREATE ROUTINE WILL PRINT A 3-DIGIT HEX ADDRESS WHICH REPRESENTS
1930  REM***THE PROM ADDRESS. VALNTER THE BYTE TO OCCUPY THAT ADDRESS IN HEX.
1940  REM***CORRECTIONS CAN BE MADE BY PRESSING THE DELETE KEY.
1950  REM***CATALOGHEN THE ENTRY IS CORRECT.PRESS VALDCLOSEAPPENDVALBACKUP
                                                                    TO RE
1969  REM***ATNFTER THE LAST ENTRY, PRESS THE 'Q' KEY TO GO TO THE PROMPRO"
1970  REM***SOFTWARE DRIVER ROUTINE
1980  REM***RESETING THE BUFFER PLACES $00 IN ALL BUFFER LOCATIONS FROM $C000
1990  REM***TO$C400 (1GO) DSAVEP TO 4K CAN BE USED FOR BUFFER
2000  PRINTCHR$(147):PRINT"SELECT MODE"
2001  PRINT"  [C]REATE HEX FILE"
2002  PRINT"  [R]ESET BUFFER"
2003  GETM$:IFM$=""GOTO2003
2004  IFM$="C"GOTO2009
2005  IFM$="R"THENZZ=0:GOTO2400
2008  GOTO2003
2009  PRINTCHR$(147);"CREATE HEXADECIMAL OBJECT FILE":PRINTCHR$(13)
2010  MA=0:BA=49152
2030  GOSUB3000
2035  PRINTAD$;" -";
2040  GETA$:IFA$=""GOTO2040
2050  A=ASC(A$)
2060  IFA>47ANDA<58THEN2090
2070  IFA>64ANDA<71THEN2090
2080  IFA=81GOTO2300
2085  GOTO2040
2090  PRINTA$;
2110  GETB$:IFB$=""GOTO2110
2120  B=ASC(B$)
2130  IFB>47ANDB<58THEN2180
2140  IFB>64ANDB<71THEN2180
2150  IFB=20THENPRINTCHR$(20);
2160  IFB=20GOTO2040
2170  GOTO2110
2180  PRINTB$;
2185  GETC$:IFC$=""GOTO2185
2186  C=ASC(C$):IFC=13GOTO2189
2187  IFC<>20GOTO2185
2188  PRINTCHR$(20);:GOTO2110
2189  PRINT"",
2190  IFA>47ANDA<58THENA1=A-48
2200  IFA>64ANDA<71THENA1=A-55
2210  IFB>47ANDB<58THENB1=B-48
2220  IFB>64ANDB<71THENB1=B-55
2250  D=A1*16+B1:POKEBA,D:BA=BA+1:MA=MA+1:GOTO2030
2300  BA=BA-1:PRINTCHR$(13):PRINT"INITIALIZING SOFTWARE DRIVER":GOTO6
2400  PRINTCHR$(147):PRINT"CLEARING 1K BUFFER;CHR$(13):PRINT"PLEASE STAND BY"
2410  FORX=4915TO49152+1024
2420  PRINTX;:FORY=1TO7:PRINTCHR$(157);:NEXTY:POKEX,0
```

```
877   IFH2>=16THENH2=H2-16:W2=W2+1:GOTO877
880   IFH2<16THENH3=H2
881   IFJ2=5THENRETURN
882   N1=N1+N
884   IFN>15THENN=N-16:L=L+1:GOTO884
885   IFN<16THEN890
890   F=F+1:H=F
891   J1=J1+1:IFJ1=1THENA5=H-1:GOSUB350:GOSUB900
892   GOSUB950
893   W=0:W1=0:W2=0:H3=0:L=0:IFJ2=5THENRETURN
895   IFJ1=16THENJ1=0:RETURN
898   GOTO804
900   Z=0:Z1=W1:Z2=W2:Z3=H3
905   A$=CHR$(A%(Z)):GOSUB600
910   A$=CHR$(A%(Z1)):GOSUB600
915   A$=CHR$(A%(Z2)):GOSUB600
920   A$=CHR$(A%(Z3)):GOSUB600
925   RETURN
950   F=F-1:F5=F+15
952   FORI=FTOF5
954   F6=PEEK(I):F7=F7+F6
955   A$CHR$(E%(F6)):GOSUB600
960   NEXTI
961   N1=F7:F7=0
962   JI=16:F=F5+1:H=F:RETURN
965   POKEC2,A%(N):C2=C2+1
980   RETURN
985   A$=CHR$(A%(T-W2)):GOSUB600
990   A$=CHR$(A%(T-H3)):GOSUB600
995   RETURN
997   REM UPLOAD ROUTINE
1000  PRINT"CHECKING PROMPRO-7 MEMORY"
1001  FORI=H5TOY5:H6=H5
1002  GET#5,A$:IFA$=""THEN1002
1003  IFAS<>"S"THEN1002
1004  FORX=1TO7:GOSUB1075:NEXTX
1005  FORJ=1TO16
1008  L1=0
1010  FORL=1TO2:GOSUB1075
1015  IFL1=1THEN1030
1020  L2=(F%(ASC(A$)))
1025  L1=1:NEXTL
1030  L9=(A%(ASC(A$)))
1033  L1=0:NEXTL
1035  L4=L2+L9
1040  Q5=PEEK(H5)
1042  REM IF ERROE THEN GO WAIT FOR NEXT S FILE LINE
1045  IFQ5<>L4THENPRINT"ERROR IN"H5:H6=H6+16:H5=H6:GOTO1060
1050  H5=H5+1:NEXTJ
1060  IFH5>=Y5THENPRINT"MEMORY OK":GOTO27
1065  NEXTI
1075  GET#5,A$:IFA$=""THEN1075
```

```
2430  NEXTX
2435  IFZZ=1GOTO2450
2440  BA=49152:PRINTCHR$(13):PRINT"INITIALIZING SOFTWARE DRIVER":GOTO6
2450  RETURN
2500  PRINTCHR$(147):PRINT"ENTER PROM ROUTINE END ADDRESS IN DECIMAL"
2510  INPUTEA
2520  BA=49152+EA:Y=BA:Y5=Y
2530  PRINT"CLEARING 1K BUFFER - STAND BY"
2540  ZZ=1:GOSUB2410
2550  GOTO270
3000  REM CONVERTS DECIMAL MEMORY ADDRESS TO 3 DIGIT HEX
3010  IFMA<16GOTO3110
3020  IFMA<256GOTO3080
3030  N1=INT(MA/256):D=N1:GOSUB3150
3050  N1$=CHR$(H):NT=MA-(N1*256)
3060  IFNT>16GOTO3090
3070  GOTO3115
3080  N1$="0":NT=MA
3090  N2=INT(NT/16):D=N2:GOSUB3150
3100  N2$=CHR$(H):N3=NT-(N2*16):GOTO3120
3110  N1$="0":N2$="0":N3=MA:GOTO3120
3115  N2$="0":N3=NT
3120  D=N3:GOSUB3150
3130  NE$=CHR$(H):AD$=N1$+N2$+N3$:RETURN
3150  IFD<10ANDD>=0THENH=D+48
3160  IFD>9ANDD<16THENH=D+55
3170  IFD>16THENPRINT"[";D;]"
3180  RETURN
READY.
```

TRS-80 COLOR COMPUTER

DOWNLOAD PROGRAM

THIS BASIC PROGRAM WILL DOWNLOAD FROM RAM (SOURCE) ADDRESSES AS SPECIFIED WITH SELECTABLE DESTINATION ADDRESS. THE "DEVICE COMMAND LINE" AS ENTERED IS SENT IN FRONT OF THE MOTOROLA FORMAT FILE (ADD "T" FOR DOWNLOAD) TO ALLOW CONTROL OF THE PROM PROGRAMMER. THE FILE ENDS WITH "S9".

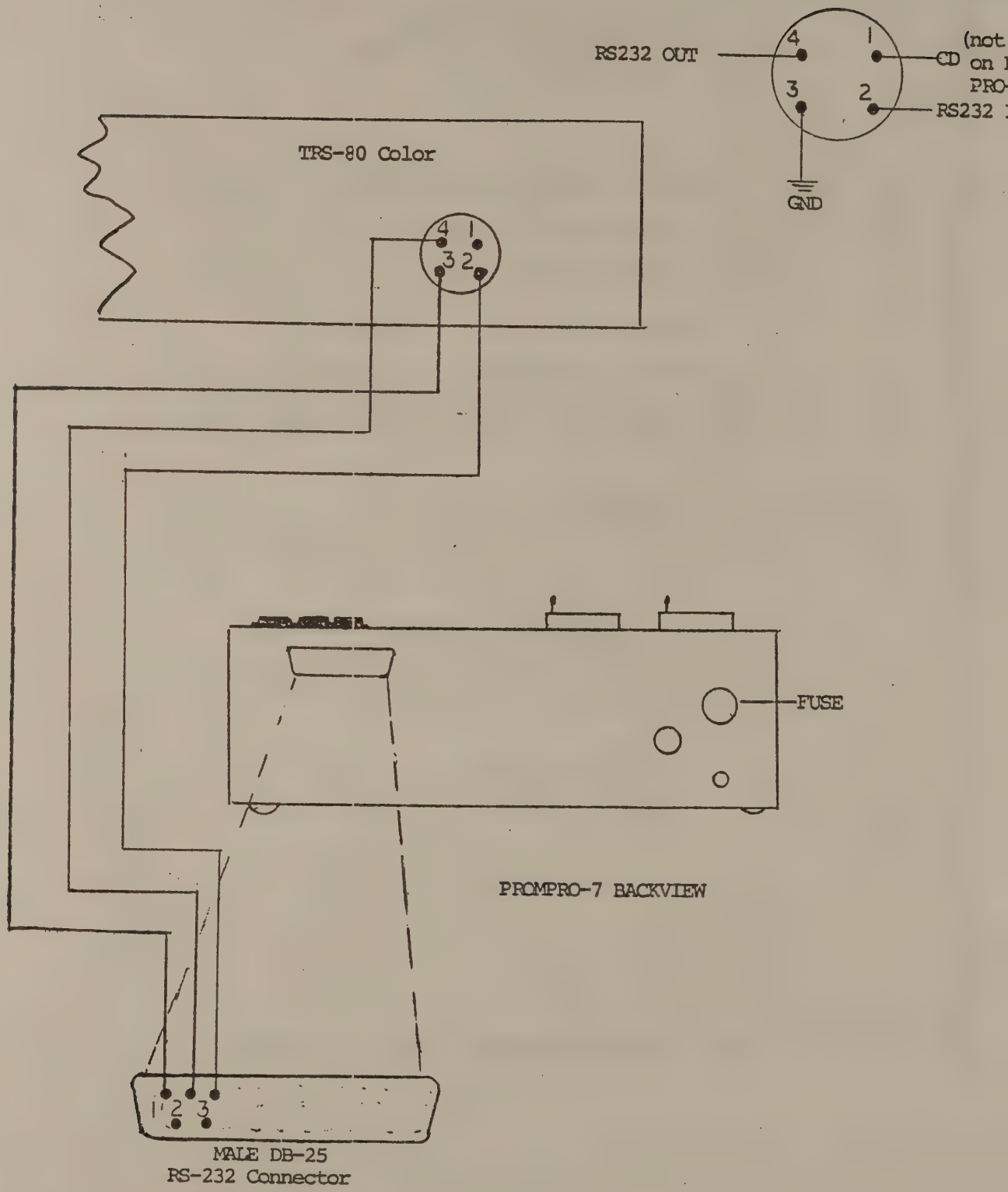
THE PROGRAM AS SUPPLIED SHOULD BE COMPATIBLE WITH EXTENDED OR NON-EXTENDED TRS-80 COLOR SYSTEMS. THE COMMENT LINES SHOW CHANGES WHICH WILL MAKE THE PROGRAM FASTER WITH EXTENDED BASIC.

THE TAPE HAS THE PROGRAM RECORDED TWICE ON ONE SIDE IN TOKEN (NORMAL) FORMAT. THIS SHOULD LOAD CORRECTLY, EVEN WITH NON-EXTENDED BASIC BUT IF IT DOESN'T, TRY THE ASCII FORMAT FILES ON THE OTHER SIDE.

TO USE YOUR COLOR COMPUTER AS A TERMINAL WITH PROMPRO-7/8, YOU CAN UTILIZE STANDARD TERMINAL PROGRAMS SUCH AS THE "SUPER COLOR" PROVIDED BY MICROWARE.

```
10 ' PROGRAM TO DOWNLOAD MEMORY FROM COLOR COMPUTER
20 ' MEMORY IN MOTOROLA HEXADECIMAL FORMAT
30 ' THIS PROGRAM IS COMPATIBLE WITH EXTENDED
40 ' OR NON-EXTENDED COLOR BASIC
50 INPUT "BAUD RATE = ";V
60 V=INT(55500/V-4.5)
70 V1=INT(V/256):POKE 149,V1
80 V=V-256*V1:POKE 150,V
90 INPUT "SOURCE STARTING ADDRESS (HEX)= ";V$
100 GOSUB 270:SS=V
110 INPUT "SOURCE ENDING ADDRESS   (HEX)= ";V$
120 GOSUB 270:SE=V
130 INPUT "DESTINATION START ADDRS (HEX)= ";V$
140 GOSUB 270:DS=V
150 INPUT "DEVICE COMMAND LINE              =";V$
160 PRINT #-2,V$
170 L=SE-SS:IF L>15 THEN L=15'DECIDE LINE LENGTH
180 IF L<0 THEN 260
190 V=L+4:CS=0:PRINT #-2,"S1";:GOSUB 390'OPEN LINE
200 V=DS:GOSUB 340'DO ADDRESS
210 FOR SS=SS TO SS+L
220 V=PEEK(SS):GOSUB 390:NEXT SS
230 V=256*INT(CS/256)-CS+255
240 GOSUB 390:PRINT #-2
250 DS=DS+L+1:GOTO 170
260 PRINT #-2,"S9":END
270 ' HEX STRING TO VALUE SUBROUTINE
280 ' FOR EXTENDED COLOR BASIC,USE INSTEAD
290 ' V=VAL("&H"+V$):RETURN
300 V=0:FOR I=1 TO LEN(V$)
310 V1=ASC(MID$(V$,I,1))-48
320 IF V1>16 THEN V1=V1-7
330 V=V*16+V1:NEXT I:RETURN
340 ' PRINT 4 HEX DIGITS SUBROUTINE
350 V1=INT(V/256):V2=V-256*V1:V=V1
360 GOSUB 390 'DO 2 DIGITS
370 V=V2:GOSUB 390 'DO OTHERS
380 RETURN
390 ' PRINT 2 HEX DIGITS SUBROUTINE
400 CS=CS+V 'DO CHECKSUM
410 ' WITH EXTENDED BASIC, NOW USE
420 ' IF V<16 THEN PRINT #-2,"0";
430 ' PRINT #-2,HEX$(V);:RETURN
440 V1=INT(V/16):V=V-16*V1
450 GOSUB 460:V1=V:RETURN
460 PRINT #-2,MID$("0123456789ABCDEF",V1+1,1);:RETURN
```

TRS-80 COLOR COMPUTER/PROMPRO-7 CONNECTION



FLEX09/PROMPRO INTERFACE SOFTWARE DRIVER

IN ORDER TO INTERFACE THE PROMPRO-7 OR PROMPRO-8 TO YOUR SWTPC OR EQUIVALANT 6809 SYSTEM YOU WILL REQUIRE A PROGRAM TO HANDLE THE SERIAL COMMUNICATIONS TO YOUR I/O PORT. THIS TYPE OF A PROGRAM IS GENERALLY REFERED TO AS A "MODEM" OR "TERMINAL" PROGRAM. SINCE BOTH THE PROMPRO-7 AND THE PROMPRO-8 ARE TERMINAL ORIENTED UNITS ALL OF THEIR FUNCTIONS CAN BE CONTROLLED VIA A TERMINAL OR A COMPUTER RUNNING UNDER THE "TERMINAL" PROGRAM. PRO78.BIN PROGRAM SUPPLIED TO YOU IS A SPECIALIZED VERSION OF THIS PROGRAM WHICH ALSO ALLOWS YOU TO TRANSFER A FILE IN THE MOTOROLA HEX FORMAT (MIKBUG PUNCH ROUTINE OR THE KANSAS CITY FORMAT). THE EXAMPLE OF THIS TYPE OF FORMAT IS PROVIDED IN THE PROGRAMMER MANUAL.
THIS SOFTWARE DRIVER UTILIZES THE MONITOR I/O DRIVER ROUTINES AS WELL AS THE PUNCH ROUTINE (THE ROUTINE THAT TAKES A SECTION OF THE MEMORY AND FORMATS IT TO THE MOTOROLA HEX FILE.) OF SWTPC SBUG-E MONITOR. IF YOU ARE NOT USING THIS MONITOR PROGRAM YOU CAN CHANGE THE ADDRESS OF THE CALLED ROUTINES TO YOUR VERSION OF THE MONITOR. THE SOURCE LISTING FOR THE PROGRAM IS PROVIDED SO THAT YOU CAN CUSTOMIZE THIS SOFTWARE.
PLEASE NOTE THAT THIS PROGRAM MUST BE LOADED IN MEMORY AND EXECUTED. TO DOWNLOAD TO THE PROGRAMMER YOU MUST TYPE "T" FOLLOWED BY THE STARTING AND ENDING ADRESS OF THE PROGRAM MEMORY.

TEKTRONIX 8002 DEVELOPMENT SYSTEM

AN EXAMPLE OF AN UP-LOAD ON THE TEKTRONIX 8002.

THE NAME OF THE FOLLOWING COMMAND IS (UPLD).

1. THE FOLLOWING LINE SPECIFIES THE INTEL HEX FORMAT.

```
PR UPLD/O PPTP 11 11
```

2. THE FOLLOWING LINE COMMANDS THE PROMPRO-7 TO SEND THE CONTENTS OF THE RAM. THIS WILL OCCUR AFTER THE 7 SECOND WAIT PERIOD.

```
PR UPLD/O PPTP 12 12
```

3. THE FOLLOWING LINE COMMANDS THE 8002 READ THE INCOMING HEX DATA FROM THE PAPER TAPE REPORT.

```
RVHEX
```

4. THE FOLLOWING LINE TERMINATES THE COMMAND FILE.

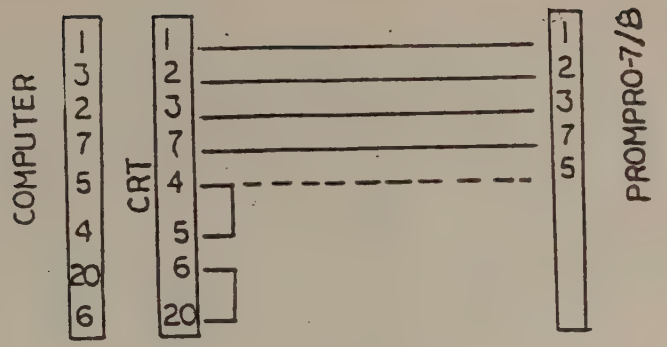
```
A UPLD
I
UO
```

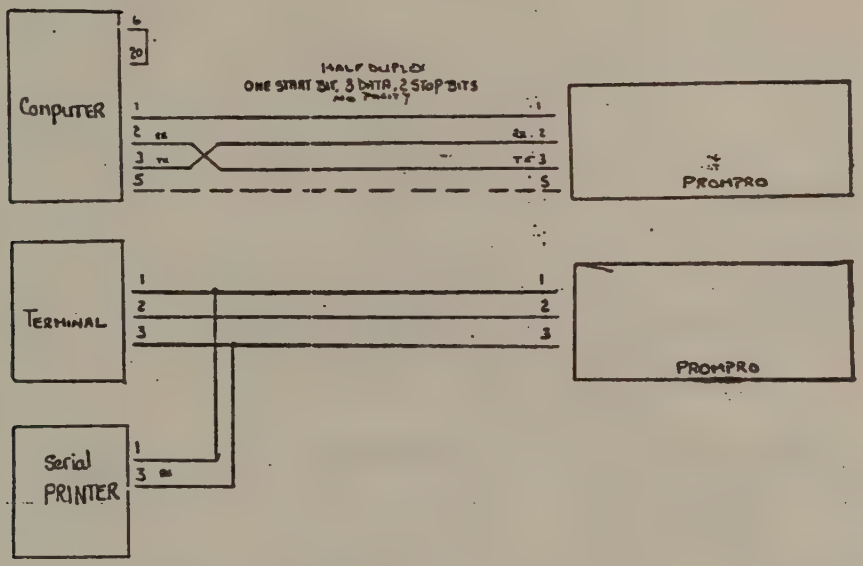
APPENDIX C

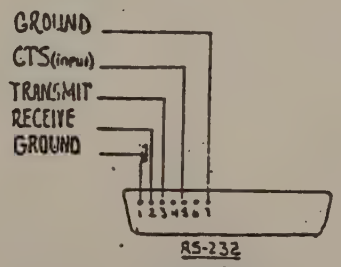
SERIAL CONNECTIONS

SERIAL RS-232C WIRING DIAGRAM.....PAGE C-1
PROMPRO-7 QUICK REFERENCE GUIDE.....PAGE C-3
OVERCOMING RS-232 INCOMPATIBILITIES.....PAGE C-4

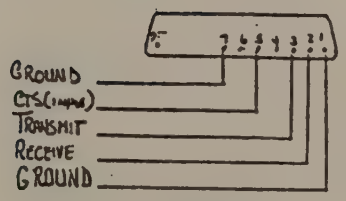
RS 232 LINK



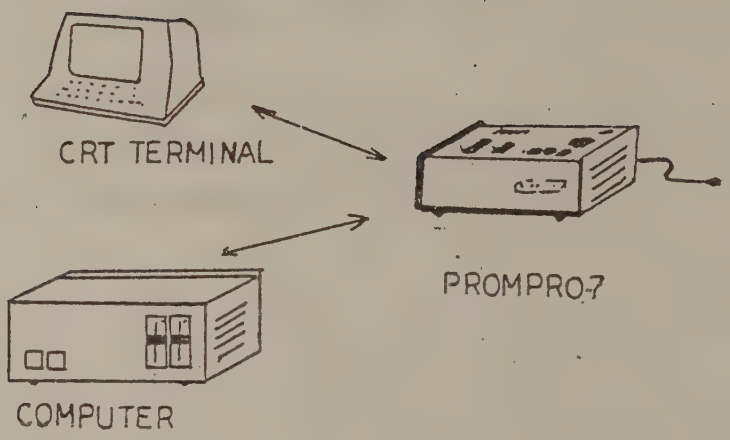




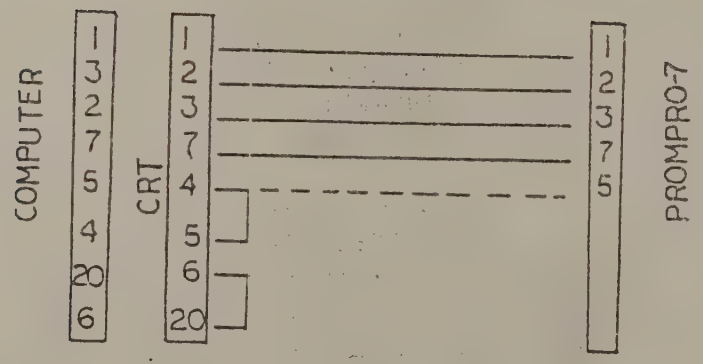
PROMPRO-7
VIEW FROM THE REAR



PROMPRO-8
RS-232
VIEWED FROM THE REAR


CRT TERMINAL
PROMPRO-7
COMPUTER

RS 232 LINK


COMPUTER
1
3
2
7
5
4
20
6
CRT
1
2
3
7
4
5
6
20
1
2
3
7
5
PROMPRO-7

PROMPRO-7 Quick Reference Guide

CONFIGURATORS

CNF #1	2716, 2516, 27C16, 8748, 8741*	CNF #6	8748H
CNF #2	2532, 25L32, 2564*	CNF #7	2764*, 27C64*
CNF #3	2732, 27C32	CNF #7	27128*, 8751*
CNF #4	2732A	CNF #8	8755*
CNF #5	879H, 8742*		

*REQUIRES EXTERNAL ADAPTER

COMMANDS

(P) PROGRAM
(V) VERIFY
(T) DOWNLOAD
(U) UPLOAD
(I) INTEL
(M) MOTOROLA
(X) EXAMINE
(*) PROGRAMMER PROMPT

RS-232 Serial Port

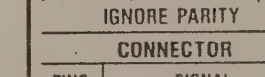
8 data bits, 1 start, 2 stop bits
IGNORE PARITY

CONNECTOR

PINS	SIGNAL
1	CHASSI GROUND
2	RECEIVE DATA
3	TRANSMIT DATA
5	CLEAR TO SEND

*CTS IS AN INPUT. LEAVE DISCONNECTED IF HARDWARE HANDSHAKE IS NOT USED.

SW 1 SW 2



BAUD RATE

ON	ON	110
ON	OFF	300
OFF	ON	1200
OFF	OFF	2400

USE UPPER CASE CHARACTER

THE SERIAL CONNECTION

How to overcome the incompatibilites of your RS232C interface.

The majority of peripheral devices on the market today such as modems, terminals, and printers are offered with a plug compatible method of interconnection known as the RS-232C interface. While this interface is one of the most standard things in the computing industry today, differences between manufacturers and device application cause incompatibilities that must be overcome in actual use. In many cases, devices can be connected with a simple cable, available from Radio Shack or your local computer store, and the devices will function without any problems. In many cases the device will go into paper weight mode and do little more than use electricity. In the second case, the user has two options; hire someone to interface the device to the machine (and at $40 to $75 an hour, this can cost as much as the device), or by a simple procedure, it can be done by the user.

THE RS-232C STANDARD

The IEEE RS-232C standardizes electrical cabling, and data protocol necessary when interconnecting two data devices. The constant electrical voltage and ampreage levels insure that all RS-232C lines can be interconnected without damaging the respective devices. The cabling standard makes these interfaces plug compatible; the data protocol insures that the data sent over the lines will contain the correct number of bits in the correct order at the correct speed.

RS-232 VOLTAGE LEVELS

The RS-232C interface has three types of lines which are signal, handshaking, and ground. The signal lines carry the data between two devices, such as a computer and a terminal, or a terminal and a modem. The ground lines insure an equipment ground, and form a circuit for the signal and handshaking lines. The handshaking lines determine if the devices are ready to transmit and receive data.

SIGNAL: The signal lines are held at approximately -16 VDC. When a data bit comes down the line the voltage is pulled to zero; in some cases it's pulled positive. Therefore, when the signal line is inactive it is negative; when data is being transmitted, the line alternates between negative and ground. For the purpose of interfacing, the most important thing to remember is that the signal lines are negative when compared to line seven, which is signal ground.

GROUND: There are two ground lines available; the signal ground line and the chassis ground line. These are both held at 0 VDC, but line seven is held at zero in respect to the signal and handshaking line. The chassis ground line connects the two devices and protects them from static and voltage biases.

HANDSHAKING: The handshaking lines are positive when ready, and ground or negative indicating a not ready condition. No signals or wave forms are sent over these lines. The simple state of either positive voltage or no voltage is the only activity on these lines. This makes the handshaking lines easy to intermix and differentiate from the signal or ground lines.

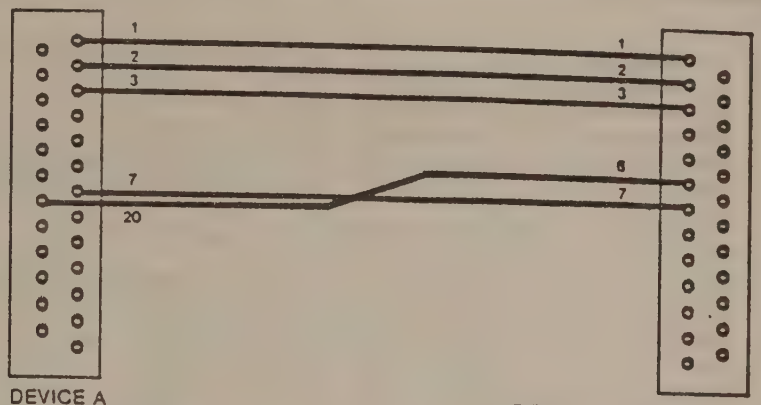
Cabling Conventions. The RS-232C cabling convention calls for the use of a standard 25 pin connector referred to as the DB-25. While the DB-25 has 25 lines available, only nine are commonly used; lines one through eight and twenty. The additional pins can be used for interfaces such as the current loop or TTL standards, which are often offered as alternatives on many devices.

The nine pins can be divided into

must be cross connected. Pin eight is connected onthe modem to pin eight on the terminal, so that the terminal knows when the modem has established the character detect. The modem still needs an input on line six to know when the terminal is ready.

Our last example deals with the 202 standard half duplex modem, which needs the handshaking lines to change anytime data is sent in either direction. Before a terminal can send data over this modem it must first bring Request To Send (RTS line four) to the active state. If the modem is ready, it will respond by bringing Clear To Send (CTS line five) high. When the terminal senses line five as being high, it will send data. The 202 modem is the only 1200 baud modem that sells for under $190.00. It is suitable for 80% of the functions done by the $600.00 Bell 212 modem, usually requiring only a more complex cable to perform correctly.

The cable that works in the largest number of applications is a direct connect cable. A direct connect cable has lines one through eight and twenty on device A connected straight across to the same pins on device B. This should be the first cable the user should try.

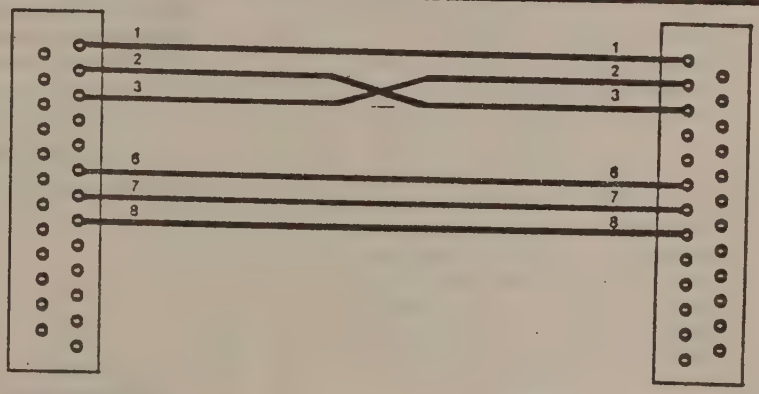


DEVICE A	
PIN	FUNCTION
1	Frame Ground
2	Send data
3	Receive data
6	no connect
7	Ground
20	Data set ready

DEVICE B	
PIN	FUNCTION
1	Frame Ground
2	Receive data
3	Send data
6	Data set ready
7	Ground
20	no connect

Figure C

The RS-232C interface often uses lines to indicate the readiness of a device to exchange information. In this example, the computer signals the modem that it is ready by bringing line twenty active. The modem's requirement is that line six be held high. Consequently, line twenty in the computer is connected to line six on the modem.



DEVICE A	
PIN	FUNCTION
1	Frame Ground
2	Send data
3	Receive data
6	Carrier detect
7	Ground
8	Data set ready

DEVICE B	
PIN	FUNCTION
1	Frame Ground
2	Send data
3	Receive data
6	Carrier detect
7	Ground
8	Data set ready

Figure D

Another modem interface involves two handshaking lines. The first line signals the modem that the terminal is ready, and the second signals the terminal that a carrier detect has been established by the modem.

DATA PROTOCOL

The data sent over the RS-232C consists of a stream of bits that, when assembled, produce a character. The amount of bits sent per character, the speed they are sent, and a few other factors make up the elements of the data protocol. These factors are baud rate, character framing, and parity.

Baud Rate: The baud rate is loosely defined as the amount of bits interchanged on an interface during one second. Of the three characteristics mentioned above, baud rate is the easiest to configure, just set both devices to the same rate.

Parity: The parity of a character is the amount of odd or even bytes it contains. The character "space" 20 hex is represented in binary as: 11000000. Because there are an even amount of bits, the space is considered to be parity even. Another popular method of determining parity is to always set the parity bit to either zero or one. Whatever method is used, make sure that both devices are set for the same parity mode.

Framing: Framing refers to the amount of data bits used to make up a character. There are four different types of bits used to make a character; start bits, stop bits, data bits and parity bits. Parity is covered in some detail in the previous paragraph.

Start and stop bits are line normal areas that surround the actual data and enable the electronics on the receiving end to tell when a character begins and ends. There is always one start bit and are one, one and one half, or two stop bits on the receiving end. With very few exceptions, setting the devices to one stop bit will work.

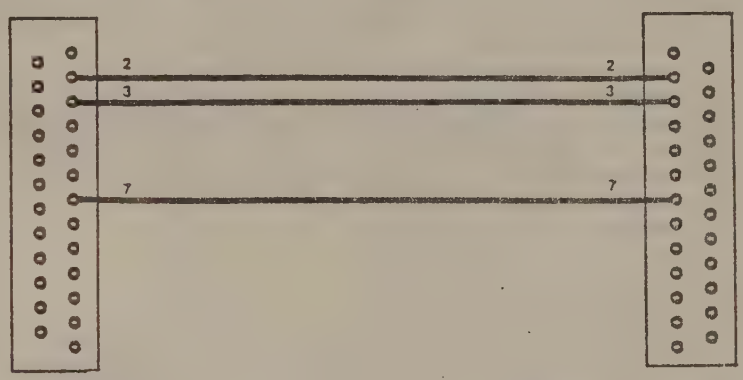
The amount of data bits used is machine dependent. Most machines, however, use eight data bits with bit

the same three groups mentioned above; the signal lines, the handshaking lines and the ground lines.These three groups are distinguished from each other by function, connector position, and electrical characteristics.

Ground. The ground lines are the easiest. When constructing a cable, pin seven is always connected to pin seven, and pin one is always connected to pin one. Pin seven is the signal ground wire and must always be connected. Pin one is the chassis ground. It's the equivalent to the third prong on a houshold electrical plug. Leaving this pin unconnected can cause disaster.

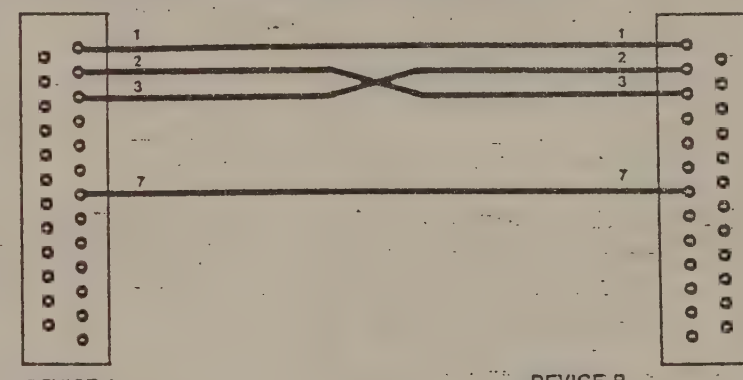
Signal. The signal lines are a bit more difficult. Pin two is transmit data and pin three is receive data. Therefore, the data transmitted by device A should be connected to the receive lines of device B, which means that these lines must often be crossed. There are three methods for determining the placement of the pins. First, consult the documentation, if pin two on both devices is transmit data, lines two and three may have to be crossed. The second method involves trial and error. Since there are only two possible combinations, if everything else is working correctly, alternating the placement should be an easy way to find which pin goes where. The last method involves the use of a tool specifically designed to solve this problem: the breakout box. Plug the breakout box into the circuit and manipulate the switches and jumpers until the indicator LED lights up for both lines two and three.

Handshaking. Aside from the ground lines, the signal lines are often the only requirement for the RS-232C interface. In other words, a cable can be constructed with only lines two, three and seven connected (and frequently is). Many devices have requirements beyond this two wire interface. Modems often require that the device connectd to it show that is connected by bringing DSR (Data Set Ready) active. Active means about +12 volts. After the modem knows that there is something connected to it, it will try to establish communications with another modem over a phone line. When this happens, the modem will bring the CD (Character Detect) line active and the device will know that is is OK to transmit and receive data over the modem link.



DEVICE A		DEVICE B	
PIN	FUNCTION	PIN	FUNCTION
2	Send data	2	Receive data
3	Receive data	3	Send data
7	Ground	7	Ground

Figure A
The RS-232C three wire interface is the minimum interface with which a device will function. It contains receive data, transmit data, and signal ground.



DEVICE A		DEVICE B	
PIN	FUNCTION	PIN	FUNCTION
1	Frame Ground	1	Frame Ground
2	Send data	2	Receive data
3	Receive data	3	Send data
7	Ground	7	Ground

Figure B
The three wire cross-over cable is a frequent modification needed to hook up devices with non-compatible pin connections.

As a more specific example, let's say we have a computer connected to a modem and that line two is transmit data on the computer and line two is receive data on the modem, and they are connected. By the same token, line three is transmit data on the modem and line three is receive data on the computer. When the wall power is turned on to both devices, the computer causes line twenty (DSR) to become active. DSR on the modem end is line six. When we connect line twenty on the computer to line six on the modem, the modem knows that the computer is connected.

When the modem sets up a carrier detect link, it raises line eight to an active state. Line eight on the modem is connected to line eight on the computer and lets the computer know that the phone connection is valid. A valid modem-computer link is now made.

Another case might involve a modem and a terminal. In this case, both the terminal and the modem use line two as transmit data, and both use line three as receive data. These lines

eight representing parity. Here again, the amount of data bits between the two devices should match.

HOW TO TROUBLE SHOOT YOUR INTERFACE PROBLEM

The RS-232C is an indestructible interface. Short of plugging it into a wall socket, the equipment probably won't be damaged by experimenting. If it doesn't work, it can be played with until it does work. Here are some helpful hints.

Problem: The power is on, the devices are connected, but the printer won't print.

Solution: There is probably a cabling problem. Switch lines two and three. If that doesn't work, switch lines four and five. If you're still not successful, short lines six, eight and twenty together.

Problem: The printer prints out the correct number of characters, but half or all of them are not what they are supposed to be.

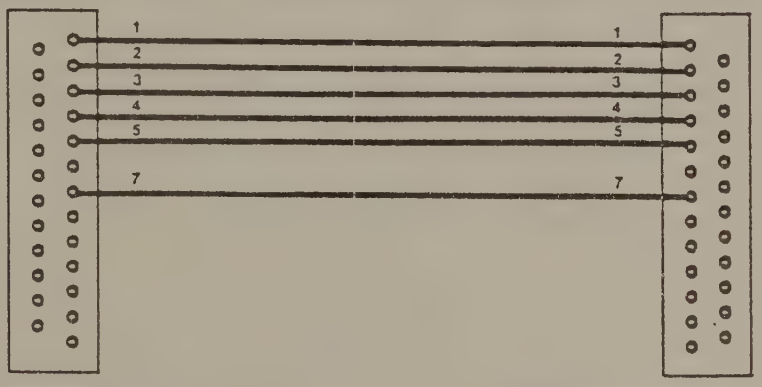
Solution: The parity bits aren't properly set.

Problem: The printer prints out garbage and the number of garbage characters doesn't match the number of characters sent to the printer.

Solution: The baud rate is incorrect.

CONCLUSION

A common quality among data processors is persistence. Most software people will work on a sofware bug for a week and never complain. Well, maybe once. . . . That same type of persistence is often needed to overcome RS-232C problems. They won't always work the first time, but they will always work.



DEVICE A		DEVICE B	
PIN	FUNCTION	PIN	FUNCTION
1	Frame Ground	1	Frame Ground
2	Send data	2	Receive data
3	Receive data	3	Send data
4	Request to send	4	Request to send
5	Clear to send	5	Clear to send
7	Ground	7	Ground

Figure E

The 202 protocol modem can only transmit data in one direction so the device it is connected to must bring line four active when it wants to transmit and wait for line five to become active.

APPENDIX D

EPROMS

LEARNING ABOUT EPROMS.....PAGE D-1
EPROM MANUFACTURER GUIDE.....PAGE D-4
EPROM PINOUT GUIDE.....PAGE D-6

GENERAL EPROM/EPROM PROGRAMMING HANDBOOK

WHAT ARE EPROMs ?

THIS QUESTION MAY BE A VERY OBVIOUS TO ALMOST ANYONE WHO HAS PURCHASED A PROGRAMMER. HOWEVER FOR THE VERY RECENT TO THE FIELD, EPROMs ARE MOS SEMICONDUCTOR MEMORIES USED MOSTLY IN CONJUNCTION WITH A MICROPROCESSOR. THE INSTRUCTIONS AND DATA FOR THE MICRO (Z-80 ,8080, 6800 AS EXAMPLES) ARE STORED PERMANENTLY IN THE EPROM. EPROM IS THE ACRONYM FOR ERASABLE PROGRAMMABLE READ ONLY MEMORY. ONCE A UV EPROM HAS BEEN PROGRAMMED IT CAN ONLY BE ERASED BY AN ULTRAVIOLET SOURCE(LOGICAL DEVICES QUV-T8 MODEL OR EQUIVALENT). FOR THIS REASON MOST COMPUTERS USE EPROMs TO STORE VITAL COMPUTER INSTRUCTIONS THAT CANNOT BE LOST AFTER POWER-DOWN.

SINCE EPROMs CANNOT BE WRITTEN TO BY THE MICROPROCESSOR DIRECTLY THE COMPUTER SYSTEM WILL NEED RAM AND DISK MEMORY FOR TEMPORARY STORAGE AS WELL AS APPLICATION DATA AND PROGRAMS.

MANY COMPANIES USE EPROMs TO STORE APPLICATION PROGRAMS SUCH AS GAMES OR COMPUTER UTILITY PROGRAMS ON EPROMS PACKAGED IN CARTRIDGES. THESE EPROMs CONTAIN MACHINE LANGUAGE INSTRUCTIONS FOR THE PARTICULAR MICROPROCESSOR USED IN THE COMPUTER. USER OFTEN NEED NOT TO BE CONCERNED WITH THE CONTENTS OF THESE EPROMs. HOWEVER FOR THE TECHNICALY INCLINED WHO WANTS TO DECIPHER THE CODE AND MAKE HIS VERSION OF THE PROGRAM THE EPROM MUST BE READ INTO AN EPROM PROGRAMMER (SUCH AS THE LOGICAL DEVICES PROMPRO SERIES). ONCE THE EPROM HAS BEEN READ INTO THE PROGRAMMER, THE DATA CAN BE SENT-UP TO A COMPUTER FROM THE SERIAL PORT FOR STORAGE ON THE DISK DRIVE.

AT THIS POINT USER MUST BE VERY FAMILIAR WITH THE LANGUAGE OF THE MICROPROCESSOR. THE UPLOADED FILE CAN BE DISASSEMBLED BY A SOFTWARE DEVELOPMENT TOOL CALLED A "DISASSEMBLER". THIS PROCESS CONVERTS THE HEX MACHINE INSTRUCTIONS TO A SERIES OF TEXT INSTRUCTION MNEMONICS REPRESENTING THE PROGRAM CODE. THIS ASSEMBLED CODE CAN BE ENTERED IN A TEXT EDITOR OR A WORD PROCESSOR FOR ALTERATIONS. ONCE THE CHANGES HAVE BEEN MADE TO THE CODE, THE PROGRAM CAN BE CONVERTED BACK TO MACHINE CODE LEVEL BY USING A SOFTWARE DEVELOPMENT TOOL CALLED AN "ASSEMBLER". THE OUTPUT OF THE ASSEMBLER THAT IS GENERALLY EITHER IN AN ABSOLUTE HEX OR IN A FORMATTED HEX FORM IS STORED BACK ON THE DISK. THIS FILE NOW IS TO BE SENT BACK TO THE PROGRAMMER BY AN OPERATION CALLED "DOWNLOAD". THE EPROM PROGRAMMER GENERALLY ACCEPTS A FILE IN A FORMATTED HEX FORM. THERE ARE TWO INDUSTRY STANDARDS; INTEL HEX OR MOTOROLA HEX. BOTH FORMATS ARE SIMILAR.(REFER TO SECTION XXX FOR MORE DETAILED DESCRIPTION). AFTER THE PROMPRO PROGRAMMER RECEIVES THE ASCII HEX FILE IT AUTOMATICALLY EXTRACTS THE DATA SECTION(MACHINE HEX CODE) AND STORES IT IN ITS RAM BUFFER. A BLANK EPROM (ONE THAT CONTAINS ALL "1" PATTERN") IS PLACED ON THE PROGRAMMER SOCKET AND THE RAM DATA IS PROGRAMMED ON THE EPROM. AFTER SUCCESSFUL PROGRAM OPERATION

THE EPROM IS REMOVED AND PLACED EITHER IN THE CARTRIDGE OR THE EPROM SOCKET INSIDE OF THE COMPUTER. HOPEFULLY THAT YOU DID NOT MAKE ANY MISTAKES IN WRITTING YOUR PROGRAM, AND AFTER YOU HAVE POWERED UP YOUR COMPUTER EVERYTHING WORKS TO YOUR SATISFACTION.
YOU MUST MAKE SURE THAT ANYTIME THAT YOU DEAL WITH ANOTHER COMPANIES SOFTWARE PROGRAM THAT YOU UNDERSTAND ALL OF THE COPYRIGHT LAWS AND BY ALTERING THOSE PROGRAMS YOU ARE NOT VIOLATING ANY OF THOSE LAWS.
EPROMS ARE FABRICATED IN A VARIETY OF MEMORY SIZES HOWEVER OFTEN IN AN 8-BIT ORGANIZATION. THE STANDARD MEMORY SIZES AVAILABLE TODAY ARE THE FOLLOWING;

GENERIC TYPE	MEMORY SIZE	PACKAGE
2716	2048 X 8	24 PIN
2732	4096 X 8	24 PIN
2764	8192 X 8	28 PIN
27128	16384 X 8	28 PIN
27256	32768 X 8	28 PIN

EPROMS ALSO ARE AVAILABLE WITH SEVERAL CHOICES OF ACCESS-TIME SPEEDS. THE DASH NUMBER AFTER THE GENERIC TYPE DESIGNATION ON THE EPROM OFTEN INDICATES THE SPEED. YOU MUST CONSULT THE DATA SHEET FOR THE PARTICULAR MANUFACTURER TO INSURE THAT THE EPROM YOU ARE USING MEETS THE ACCESS-TIME REQUIRED BY YOUR COMPUTER.
THE PROMPRO EPROM PROGRAMMING EQUIPMENT IS INDIFFERENT TO THE EPROM ACCESS-TIME AND CAN PROGRAM ANY SPEED EPROM.

EPROMS CAN BE ERASED AND REPROGRAMMED MANY TIMES, ALTHOUGH AS THE NUMBER OF ERASER TIME INCREASES SO DOES THE TIME REQUIRED TO ERASE THEM. A NEW EPROM WITH A GOOD QUALITY QUARTZ WINDOW CAN TAKE AS LITTLE AS 7 MINUTES TO ERASE WITH A QUV-T8/Z EPROM ERASER.

MOST OF THE EPROMS TODAY OPERATE WITH A POWER SUPPLY VOLTAGE (VCC) OF 4.75 - 5.25 WITH 5.00 NOMINAL VALUE. VCC IS APPLIED TO PIN 24 FOR 24 PIN PACKAGES AND PIN 28 FOR 28 PIN PACKAGES.
IN ORDER TO PROGRAM AN EPROM IT IS REQUIRED TO APPLY A HIGH VOLTAGE TO ONE OF THE SIGNAL PINS(PROG) AND PULSE ONE OTHER OR THE SAME PIN FOR A PERIOD OF 50 MILLISECONDS, WHILE APPLYING THE PROPER ADDRESS AND DATA TO THE EPROM.

EPROMS ARE EXTREMELY SENSITIVE TO STATIC ELECTRICITY. IT IS IMPERATIVE THAT YOU ALWAYS USE CONDUCTIVE FOAM PAD WHEN TRANSPORTING EPROMS (LOGICAL PN# AP-10). YOU MUST ALSO USE A GROUNDING STRAP OR GROUNDING BAR WHENEVER WORKING DIRECTLY WITH EPROMS OR THE PROGRAMMING EQUIPMENT.

MOST EPROMS WHEN SHIPPED FROM THE FACTORY ARE ALREADY ERASED (ALL "FF PATTERN). IT IS A GOOD IDEA HOWEVER TO CHECK EPROMS PRIOR TO PROGRAMMING FOR THE BLANK PATTERN.

EPROMs ARE ALSO VERY SENSITIVE TO POWER GLITHES. IF YOU ARE HAVE A SEVERE STORM IN THE AREA OR UNUSUALLY HIGH POWER INTERRUPTIONS IT IS BEST TO POSTPONE PROM PORGRAMMING UNTIL THE PROBLEM IS CLEARED. REMOVE YOUR EPROMs FROM THE EQUIPMENT DURING POWER-UP AND POWER-DOWN.

FADING. FADING OCCURS ON EPROMs THAT ARE POORLY PROGRAMMED OR ARE DEFECTIVE. EPROMs REQUIRE A CERTAIN MINIMUM NUMBER OF ELECTRICAL CHARGES IN THE FLOATING GATE OF THE FET SWITCHES. THESE CHARGES TEND TO LEAK OUT GRADUALLY OVER A LONG PERIOD OF TIME. IF THE EPROM PROGRAMMING EQUIPMENT DOES NOT SUPPLY THE REQUIRED AMOUNT OF CHARGES, THE EPROM MAY INITIALY PROGRAM AND VERIFY PROPERLY AND WORK IN YOUR COMPUTER FOR SEVERAL MONTHS, UNTIL THE EPROM BITS START CHANGING STATE INTERMITTENTLY. SOME MANUFACTURERS OF EPROM PROGRAMMING EQUIPMENT MAY PROMISE FAST PROGRAM TIME ON EPROMs TO SELL THEIR EQUIPMENT, WITHOUT CONSIDERATIONS TO THE PROPER PROGRAMMING ALGORITHMS. LOGICAL DEVICES INC. ADHERES VERY STRONGLY TO THE MANUFACTURERS SPECIFICATIONS TO MAXIMIZE THE FIELD RELIABILITY AND LONGEVITY OF YOUR EPROMs.

EPROM TYPES. WITHIN THE SAME GENERIC FAMILY THERE MAY BE SEVERAL TYPE OF EPROMs THAT VARY AS FAR AS THE PROGRAMMING VOLTAGE REQUIREMENTS. THERE ARE PRESENTLY TWO COMMON PROGRAMMING VOLTAGE LEVELS; 25 VOLTS AND 21 VOLTS. SOME NEW DEVICES WILL REQUIRE 12.5 PROGRAMMING VOLTAGE.
FOR EXAMPLE A 2732 EPROMs REQUIRES A 25 VOLT PROGRAM VOLTAGE WHEREAS, A 2732A REQUIRES 21 VOLTS TO PROGRAM. ATTEMPTING TO PROGRAM A 2732A WITH 25 VOLT CAN DESTROY THE DEVICE.
EPROMs HAVE A MUCH BETTER PROGRAMMING YIELD THAN BIPOLAR PROMs. USER MUST HAVE BETTER THAT 5% PROGRAMMING YIELD FOR EPROMs. IF THE FAILURE RATES EXCCEED 5% USER MUST SUSPECT THE EPROMs OR THE EQUIPMENT.
WHERE CAN YOU BUY EPROMs?. IT IS RECOMMENDED THAT YOU SELECT RELIABLE SOURCES FOR YOUR COMPONENTS. AVOID BUYING EPROMs FROM UNCERTAIN ORIGINS OR FROM A QUESTIONABLE SURPLUS STORE. LOGICAL DEVICES INC. CAN RECOMMEND THE FOLLOWING SOURCES FOR YOUR EPROMs;

CASPIAN TECHNOLOGY
781 W. OAKLAND PRK BLVD.
SUITE 580
FT. LAUDERDALE FL 33311'
TEL; 305-974-0977

HAMILTON AVNET
HALLMARK
ARROW ELECTRONICS
HAMMOND ELECTRONICS

PLEASE CHECK YOUR LOCAL TELEPHONE BOOK FOR
LOCAL BRANCH OFFICE OF THE ABOVE DISTRIBUTORS.

MANUFACTURERS OF PROGRAMMABLE READ-ONLY MEMORIES

ADVANCED MICRO DEVICES
901 THOMPSON PL
SUNNYVALE, CA 94086
(408) 732-2400

AMERICAN MICROSYSTEMS INC
3800 HOMESTEAD RD
SANTA CLARA, CA 95051
(408) 246-0330

ELECTRONIC ARRAYS
550 E MIDDLEFIELD RD
MT VIEW, CA 94043
(415) 964-4321

FAIRCHILD SEMICONDUCTOR
464 ELLIS ST
MT VIEW, CA 94042
(415) 962-5011

FUJITSU MICROELECTRONICS
2945 OAKMEAD VILLAGE CT
SANTA CLARA, CA 95051
(408) 729-1700

GEN INSTRUMENT CORP
MICROELECTRONIC DIV
600 W JOHNS ST
HICKSVILLE, NY 11802
(516) 733-3000

HARRIS SEMICONDUCTOR
BOX 883
MELBOURNE, FL 32901
(305) 724-7407

HITACHI AMERICA LTD
1800 BERING DR
SAN JOSE, CA 95112
(408) 292-6404

HUGHES AIRCRAFT CO
SOLID ST PROD DIV
500 SUPERIOR AVE
NEWPORT BCH, CA 92663
(714) 759-2411

INTEL CORP
3065 BOWERS AVE
SANTA CLARA, CA 95051

INTERSIL INC
10710 N TANTAU AVE
CUPERTINO, CA 95014

MOSTEK CORP
1215 W CROSBY DR
CARROLLTON, TX 75006

MITSUBISHI/MEICO
3030 E VICTORIA ST
COMPTON, CA 90221
(213) 537-7131

MONOLITHIC MEMORIES INC
1165 E ARQUES AVE
SUNNYVALE, CA 94086
(408) 739-3535

MOTOROLA INC
INTERGRATED CIRC DIV
3501 ED BLUESTEIN BLV
AUSTIN, TX 78721
(512) 928-6000

NATIONAL SEMICONDUCTOR
2900 SEMICONDUCTOR DR
SANTA CLARA, CA 95051
(408) 737-5000

NEC MICROCOMPUTERS INC
173 WORCESTER ST
WELLESLEY, MA 02181
(617) 239-1910

NITRON INC
10420 BUBB RD
CUPERTINO, CA 95014
(408) 255-7550

OKI SEMICONDUCTOR
SUITE 405
SANTA CLARA, CA 95051
(408) 984-4842

PLESSEY SEMICONDUCTOR
1641 KAISER AVE
IRVINE, CA 92714
(714) 540-9979

PANASONIC
1 PANASONIC WAY
SECAUCUS, NJ 07094
(201) 348-7276

RAYTHEON SEMICONDUCTOR
350 ELLIS ST
MT VIEW, CA 94042
(415) 968-9211

SGS-ATES SEMICONDUCTOR CRP
240 BEAR HILL RD
WALTHAM, MA 02154
(617) 890-6688

RCA SOLID STATE DIV
RTE 202
SOMERVILLE, NJ 08876
(201) 685-6000

SIGNETICS CORP
811 E ARQUES AVE
SUNNYVALE, CA 94086
(408) 739-7000

TEXAS INSTRUMENTS INC
BOX 225012, M/S 308
DALLAS, TX 75265

SYNERTEK
BOX 552
SANTA CLARA, CA 9505
(408) 988-5611

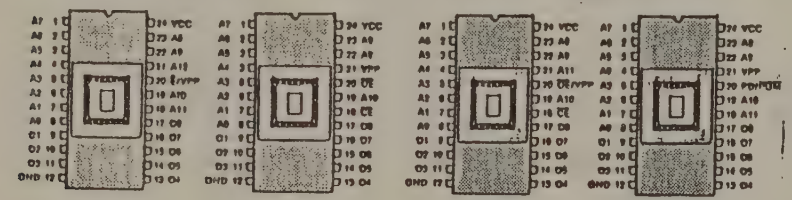
TOSHIBA AMERICA INC
2151 MICHELSON DR
IRVINE, CA 92715
(714) 955-1155

XICOR INC
1221 INNSBRUCK DR
SUNNYVALE, CA 94086
(408) 734-3041

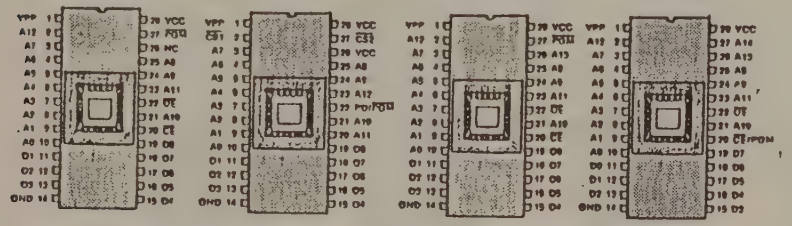
ZILOG
10460 BUBB RD
SUNNYVALE, CA 94086
(408) 446-4666

EPROM PIN-OUTS

MCM 68764, 68766 | 2516, 2716, 27C16 | 2732, 27C32, 2732A | 25L32, 2532



2764, 27C64, 2764A | 2564 | 27128, 27128A | 27256



LOGICAL DEVICES INC.

EPROM PIN-OUT GUIDE

EPROM TYPE	VCC PIN	GRND PIN	VPP PIN	VPP VOLTAGE	PGM PULSE	PGM WITH PULSE	PIN OUTPUT ENABLE
2716	24/26	12/14	21/23	25V ± 1V	18/20	50ms	20/22
2732	24/26	12/14	20/23	25V ± 1V	20/22	50ms	20/22
2732A	24/26	12/14	20/22	21V ± 1V	20/22	50ms	20/22
2764	28	14	1	21V ± v	27	50ms	20/22
2564	28	14	1	21V ± V	22	50ms	20/22
27128	28	14	1	21V ± V	27	50ms	20/22
27256	28	14	1	12.5V ± 5V	27	*	20/22
68764	24/26	12/14	20/22	25V ± 1V	20/22	*	20/22
2532	24/26	12/14	20/22	25 ± 1V	20/25	50ms	20/22

PIN NUMBERS ARE BOTH INDICATED FOR BOTH 24 PIN AND 28 PIN SOCKETS BY THE "/".

2764A, 27128A 12.5 VPP
VPP = PROGRAM VOLTAGE
VCC = SUPPLY CURRENT 5V ± .25V

LEADER IN PROGRAMMING EQUIPMENT:

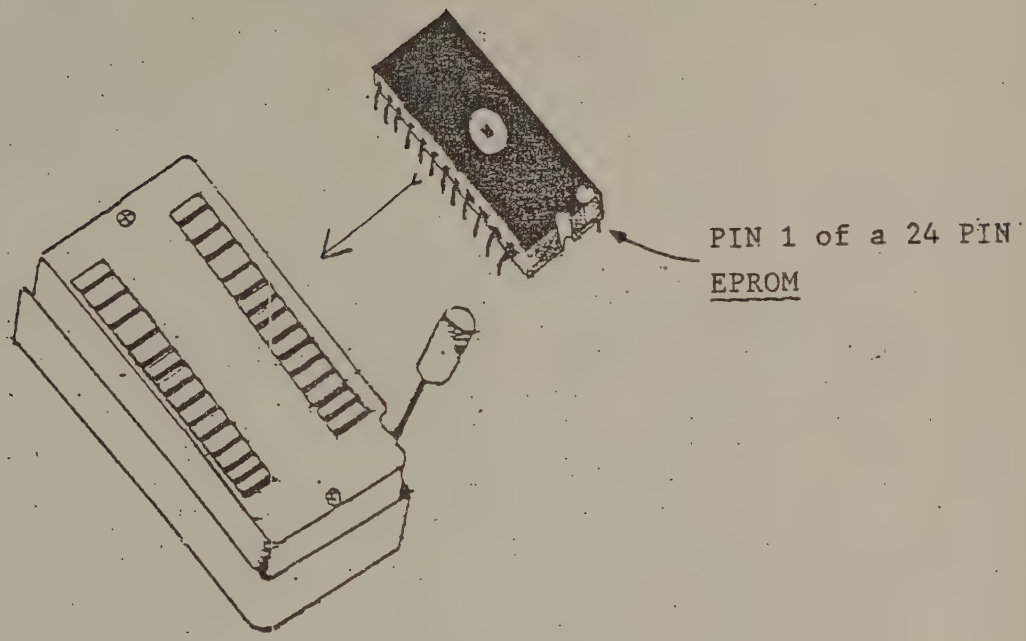
LOGICAL DEVICES INC.

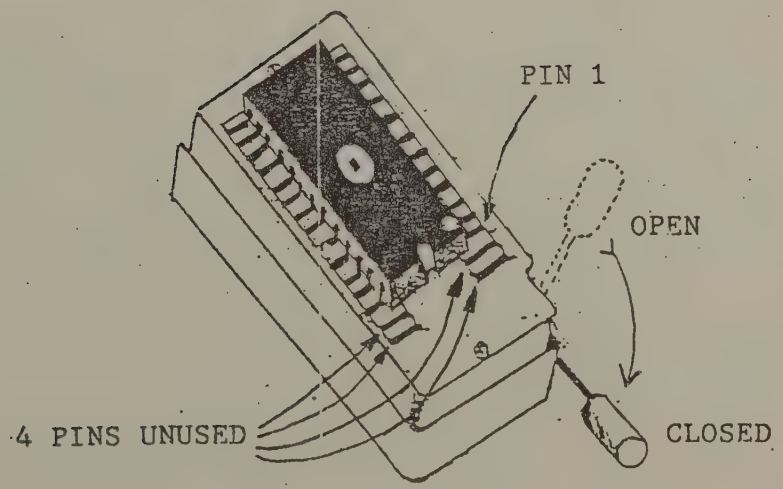
APPENDIX E

HOW TO INSERT EPROMS IN THE ZIF SOCKETS

INSERTING 24 PIN DEVICE IN 28 PIN SOCKET DIAGRAM....PAGE E-1
28 PIN ZIF SOCKET DIAGRAM.....PAGE E-1

INSERTING A 24 PIN DEVICE IN SOCKET 1





ZIF SOCKET 1

APPENDIX F

XP7 FIRMWARE OPTION

MENU.....PAGE F-1
MENU DICTIONARY.....PAGE F-2

MENU
******

(I) :SPECIFY INTEL HEX

(K) :SWAP 0000-07FF AND 0800-0FFF

(L) :TRANSMIT 256 BLOCK IN ASCII

(M) :SPECIFY MOTOROLA HEX

(N) :BLANK CHECK

(P) :PROGRAM EPROM

(T) :DOWN-LOAD FROM HOST

(U) :UPLOAD TO HOST

(V) :VERIFY EPROM TO RAM

(W) :UPLOAD WITH WAIT

(X) :EXAMINE MEMORY

(Z) :SEND Z (1AH) AFTER UPLOAD

(I) :SPECIFY INTEL HEX

INPUTING AN "I" COMMANDS THE PROMPRO-7 TO TRANSMIT AND RECEIVE DATA VIA THE RS-232 CONNECTOR IN THE INTEL HEX FORMAT.

(K) :SWAP 0000-07FF AND 0800-0FFF

INPUTING A "K" COMMANDS THE PROMPRO-7 TO ALLOW 2K OF DATA TO BE SWAPPED IN MEMORY WITHOUT LOSS OF ANY DATA. THIS COMMAND IS USEFUL IN FILLING THE RAM BUFFER WITH TWO 2716'S TO PROGRAM ONE 2732 OR FILLING THE RAM BUFFER WITH ONE 2732 TO PROGRAM TWO 2716'S.

(L) :TRANSMIT 256 BLOCK IN ASCII

INPUTING AN "L" COMMANDS THE PROMPRO-7 TO UPLOAD 256 BITS OF DATA AT A TIME. TWO COLUMNS ARE GENERATED. ONE IS THE HEX DATA, THE SECOND IS THE ASCII EQUIVALENT. INPUTING ANY CHARACTERS UPLOADS THE NEXT 256 BITS. INPUTING AN ENTER OR RETURN EXITS THIS MODE.

(M) :SPECIFY MOTOROLA HEX

INPUTING AN "M" COMMANDS THE PROMPRO-7 TO TRANSMIT AND RECEIVE DATA IN THE MOTOROLA HEX FORMAT.

(N) :BLANK CHECK

INPUTING AN "N" COMMANDS THE PROMPRO-7 TO TEST AN EPROM IN EITHER SOCKET 1 OR 2 FOR AN ERASED PATTERN.

(P) :PROGRAM EPROM

INPUTING A "P" COMMANDS THE PROMPRO-7 TO BEGIN PROGRAMMING.

(T) :DOWNLOAD FROM HOST

INPUTING A "T" COMMANDS THE PROMPRO-7 TO ALLOW INCOMING DATA IN INTEL OR MOTOROLA HEX FORMAT FROM THE RS-232 CONNECTOR TO BE INPUTED INTO THE RAM BUFFER AFTER A 7 SECOND DELAY.

(U) :UPLOAD TO HOST

INPUTING A "U" FOLLOWED BY A 0, (U0), COMMANDS THE PROMPRO-7 TO TRANSMIT IN EITHER INTEL OR MOTOROLA HEX VIA THE RS-232 CONNECTOR TO A HOST SYSTEM AFTER A 5 TO 7 SECOND DELAY.

(V) :VERIFY EPROM TO RAM

INPUTING A "V" COMMANDS THE PROMPRO-7 TO VERIFY AN EPROM AGAINST THE RAM BUFFER CONTENTS.

(W) :UPLOAD WITH WAIT

INPUTING A "W" COMMANDS THE PROMPRO-7 TO TRANSMIT DATA VIA INTEL OR MOTOROLA HEX FORMAT VIA THE RS-232 CONNECTOR AFTER A 25 SECOND DELAY.

(X) :EXAMINE MEMORY

INPUTING AN "X" FOLLOWED BY THE DESIRED ADDRESS TO BE EXAMINED, (X0000), COMMANDS THE PROMPRO-7 TO OUTPUT TO THE HOST SYSTEM THAT ADDRESS ALONG WITH THE DATA STORED IN THAT LOCATION (0000;AA). THE DATA CAN BE CHANGED BY INPUTING THE NEW DATA. THE ADDRESS WILL AUTOMATICALLY INCREMENT TO THE NEXT ADDRESS. INPUTING A SPACE INCREMENTS THE ADDRESS WITHOUT ALTERING THE CONTENTS AT THAT LOCATION. A CARRIAGE RETURN EXITS THIS MODE.

(Z) :SEND^Z (1AH) AFTER UPLOAD

INPUTING A ^ Z BEFORE AN UPLOAD COMMANDS THE PROMPRO-7 TO SEND A CONTROL Z, (1AH) AFTER THE UPLOAD IS EXECUTED. THE CONTROL Z ACTS AS AN END OF FILE FOR THE HOST SYSTEM.

PROMPRO-7 REV. 2.0 4-1-84

DEAR CUSTOMER

PLEASE MAKE ANY COMMENTS THAT YOU MAY HAVE ABOUT THE MANUAL IN THE SPACE PROVIDED BELOW AND RETURN TO US AS SOON AS POSSIBLE. YOUR COMMENTS WILL BE REVIEWED AND CONSIDERED FOR THE NEWER VERSION MANUALS.
PLEASE BE SPECIFIC IN REGARDS TO YOUR COMMENTS AND SPECIFY AREAS OF THE MANUAL THAT EITHER HARD TO READ, UNCLEAR OR POORLY ORGANIZED. YOUR FEEDBACK IS GREATLY APPRECIATED.
PLEASE MAKE SURE TO FILL IN YOUR NAME AND ADDRESS BELOW SO THAT WE SEND YOU ANY FUTURE UPDATES.

COMMENTS

NAME:
COMPANY:
ADDRESS:

TELEPHONE: DATE: